UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR

Grado en Ingeniería Informática

TRABAJO FIN DE GRADO

Desarrollo de herramienta de análisis de calidad de distribución de contenidos multimedia mediante árboles multicast

Autor: Jorge Gutiérrez Díaz
Tutor: Luis De Pedro Sánchez
Ponente: Jorge Enrique López de Vergara Méndez

junio 2020

Jorge Gutiérrez Díaz
Desarrollo de herramienta de análisis de calidad de distribución de contenidos multimedia mediante árboles multicast

Jorge Gutiérrez Díaz
C\ Francisco Tomás y Valiente Nº 11

IMPRESO EN ESPAÑA – PRINTED IN SPAIN

A mi familia y amigos por hacer posible ser lo que soy hoy

La experiencia es el nombre que le damos a nuestros errores.

Oscar Wilde

Agradecimientos

En primer lugar me gustaría dar las gracias a Luis de Pedro Sánchez y a Jorge López de Vergara por la oferta de este proyecto y por el apoyo de ambos a lo largo del desarrollo de este.

Por otro lado, me gustaría agradecer a mis compañeros y amigos por los buenos momentos que he compartido con ellos durante esta etapa de mi vida.

Por último, agradecer a mi familia por su apoyo y cariño, y por esta siempre a mi lado en todo momento.

Resumen

En las últimas décadas, los sistemas de televisión sobre IP han conseguido un gran éxito gracias al desarrollo de la infraestructura de Internet. Las distintas operadoras, que ofrecen servicios por Internet, han aprovechado la oportunidad que brinda IPTV para desplegar distintas aplicaciones multimedia y poder competir con otras alternativas, como la televisión por satélite o la televisión digital. Por este motivo que se necesitan herramientas de monitorización que permitan encontrar anomalías que puedan afectar a estos servicios.

Este trabajo se centra en el desarrollo de una herramienta de monitorización de este tipo, capaz de realizar seguimientos de los servicios *live television* utilizando una distribución *multicast*, y los protocolos RTP e IGMP para su transmisión y señalización respectivamente.

La herramienta muestra si han ocurrido anomalías en el servicio y como afectan estas en el transcurso de la sesión. Por otro lado, calcula la calidad de servicio de la red y como varía a lo largo de la sesión. Por último, obtiene medias de la calidad de servicio y evalúa su resultado en la sesión.

En definitiva, el proyecto informa sobre las pérdidas de paquetes, *jitter* y anomalías IGMP de manera periódica, en intervalos de 10 segundos, con el objetivo de identificar posibles incidencias y resolverlas lo antes posible.

Por último, se han realizado distintos casos de prueba para verificar el funcionamiento del proyecto con capturas reales de tráfico.

Palabras clave

IPTV, Live Television, RTP, IGMP, calidad de servicio, pérdidas de paquetes, jitter

Abstract

In recent decades, television over IP systems have achieved great success thanks to the development of Internet infrastructure. The various operators, which offer services over the Internet, have taken advantage of the opportunity provided by IPTV to deploy different multimedia applications and to be able to compete with other alternatives, such as satellite television or digital television. This is why monitoring tools are needed to find anomalies that may affect these services.

This work focuses on the development of such a monitoring tool, capable of tracking Live television services using a multicast distribution, and RTP and IGMP protocols for transmission and signaling respectively.

The tool shows whether anomalies have occurred in the service and how they affect them over the course of the session. On the other hand, it calculates the quality of service of the network and how it varies throughout the session. Finally, you get averages of the quality of service and evaluate your result in the session.

In short, the project reports on packet losses, jitter and IGMP anomalies on a regular basis, at 10-second intervals, with the aim of identifying potential incidents and resolving them as soon as possible.

Finally, different test cases have been carried out to verify the operation of the project with actual traffic captures.

Keywords

IPTV, Live Television, RTP, IGMP, quality of service, packet losses, jitter

Índice

Listas

Lista de algoritmos

Lista de códigos

Lista de cuadros

Lista de ecuaciones

Lista de figuras

Lista de tablas

Lista de cuadros

1

Introducción

En este capítulo se describe la motivación por la que se ha decidido hacer este proyecto, sus objetivos, las fases por la que ha pasado el mismo y la estructura del documento final.

1.1. Motivación del proyecto

En las últimas décadas, la forma en que la población visualiza el contenido audiovisual a sufrido grandes cambios. En especial, el método de difusión de recursos multimedia a cobrado gran importancia debido al desarrollo de los sistemas Internet Protocol Television (IPTV) [1]. Este protocolo distribuye el contenido multimedia entre los clientes de Internet Service Provider (ISP) [2] reservando parte del ancho de banda que tengan contratado los clientes para garantizar un *QoS* y *QoE* adecuado. Las operadoras han aprovechado la oportunidad de estos servicios y ofrecen la posibilidad de ver series de televisión, series o películas a sus clientes. Existen varios tipos de servicios que ofrece IPTV , pero en ese proyecto nos centramos en *live television*.

La principal característica de *live television* es que el contenido audiovisual se transmite una sola vez por un árbol *multicast* que se copia y envía a los clientes que se hayan suscrito a él. Para la subscripción y abandono del árbol de difusión se utiliza Internet Group Management Protocol (IGMP) , donde el Set-Top-Box (STB) desempeña el papel de cliente cuando se utiliza la televisión.

Dada la importancia que han adquirido este tipo de tecnologías para los diferentes ISP, es necesario desarrollar herramientas que monitoricen en tiempo real este tipo de servicio y ayuden a mantener una calidad de servicio adecuado para los clientes.

1.2. Objetivos y enfoque

El principal objetivo del proyecto es desarrollar una herramienta de monitorización del protocolo IPTV que utiliza el servicio *live television*, donde se examina los canales que estén visualizando los clientes mediante los paquetes pertenecientes a IGMP. Una vez conocido a que árbol de multidifusión

esta suscrito el cliente, se filtran los paquetes pertenecientes al árbol y se realizan las medidas adecuadas a la calidad de transmisión para detectar los posibles problemas que pueden llegar a ocurrir. Dentro de los posibles problemas que pueden ocurrir, el proyecto será capaz de calcular el porcentaje de pérdidas, *jitter*, latencia, *throughput* y anomalías IGMP, y a la vez estimar Mean Opinion Score (MOS) .

1.3. Planificación del TFG

A continuación, la figura 1.1 muestra un diagrama de Gantt para representar el tiempo dedicado a cada parte del proyecto y en la figura 1.2 un resumen del trabajo llevado a cabo durante este proyecto.

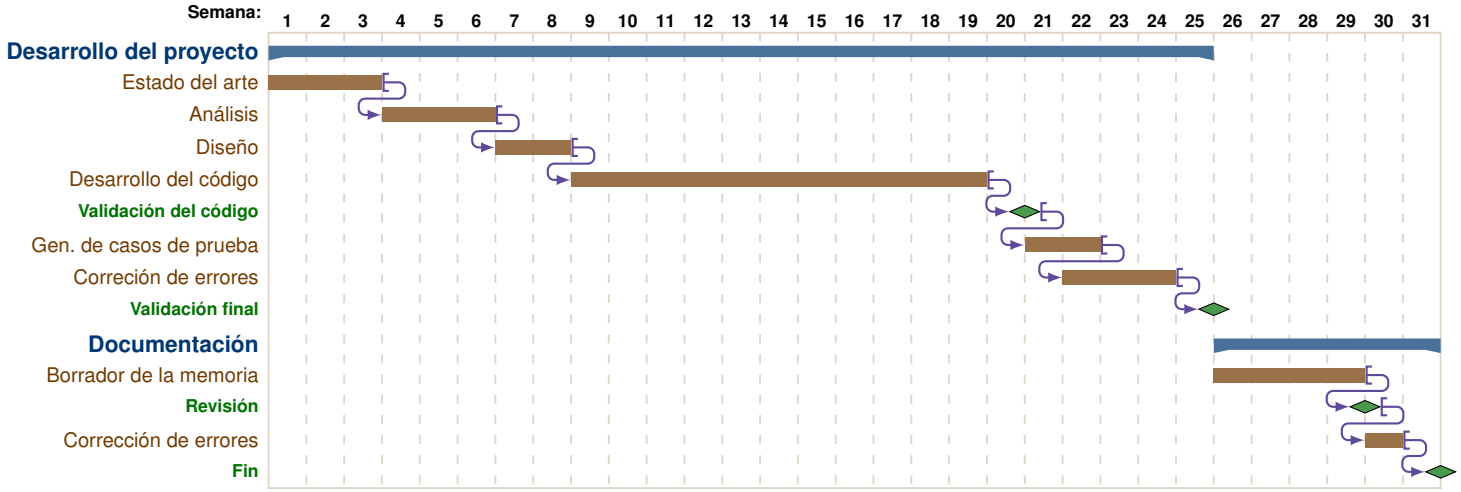


Figura 1.1: Diagrama de Gantt del trabajo realizado.

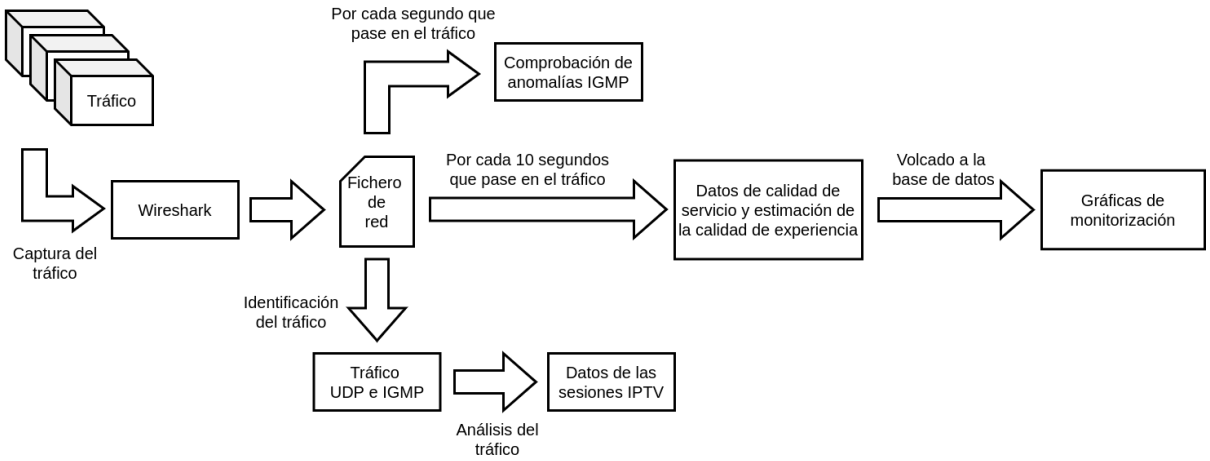


Figura 1.2: Diagrama del trabajo realizado

1.4. Estructura del documento

El documento se encuentra dividido en 7 capítulos de la siguiente manera:

Introducción Se presenta en qué consiste el Trabajo Final de Grado y qué objetivos se persiguen. De igual manera, se muestra el tiempo dedicado al proyecto, un diagrama del trabajo realizado y cómo esta dividido el documento.

Estado del arte Breve repaso de los aspectos más importantes de servicios de vídeo sobre IP en los últimos años. Además, se introduce el protocolo IGMP, el cual se ha utilizado para realizar un seguimiento sobre que árboles de multidifusión están consumiendo los clientes. Por último, se describe el protocolo Real-time Transport Protocol (RTP) en el cual se utiliza para obtener datos que utiliza la herramienta.

Análisis Se describe las funcionalidades que se pretenden conseguir con la aplicación y se enumeran los requisitos funcionales y no funcionales.

Diseño Se muestra la división de la herramienta en cuatro fases junto con un diagrama para una visión global de la misma. Además se explican ciertas decisiones tomadas para el desarrollo de la aplicación.

Desarrollo El capítulo consiste en explicar como se identifican los paquetes pertenecientes a cada árbol de difusión y que estructura se ha desarrollado para conseguirlo. Además, se presenta como obtener las medidas de calidad y su posterior representación visual.

Pruebas y resultados Se detalla cómo se han conseguidos los casos de prueba para comprobar la validación del programa y de qué tratan esos casos.

Conclusiones y trabajo futuro Se expone las conclusiones obtenidas y qué aspectos se podrían desarrollar en el futuro.

2

Estado del arte

2.1. Introducción

En el siguiente capítulo se realiza una breve introducción a los protocolos IPTV, IGMP y RTP para recordar el funcionamiento y características de estos. Además, se explica qué es la calidad de servicio y la calidad de experiencia, y cómo intervienen en nuestra herramienta.

2.2. IPTV

IPTV es la distribución de canales de televisión tradicionales, películas, textos, gráficos, datos y contenido de vídeo y audio bajo demanda sobre una red IP de banda ancha privada [1]. La parte encargada de enviar los contenidos multimedia son los servidores Super Head-end Office (SHO) los cuales codifican y fragmentan la información para enviarla a los distintos Video Head-end Office (VHO) a través de IP [2]. Los VHO se sitúan en áreas metropolitanas de la población y son los encargados de llevar el contenido a los equipos que lo solicitan. A diferencia de los métodos tradicionales, como la televisión por satélite [3], la entrega de los contenidos multimedia se realizan cuando el cliente lo desea.

IPTV proporciona diversos servicios, los cuales pueden dividirse en dos grupos:

Live television Los servidores de contenido transmiten el contenido multimedia en vivo a través de árboles de difusión utilizando *multicast*. Cuando un cliente quiera ver un canal este deberá suscribirse al nuevo árbol de difusión correspondiente y abandonar el árbol anterior en el caso de que estuviera suscrito a uno previamente.

Video on Demand (VoD) Los contenidos multimedia se encuentran almacenados en los servidores de contenido y el cliente solicita acceso a ellos. En este tipo de casos, el flujo del contenido se realiza por *unicast*.

2.3. IGMP

IGMP es el mecanismo de control utilizado para gestionar la entrega del tráfico *multicast* a los usuarios interesados y autorizados. Los comandos de IGMP hablan con los equipos de los extremos para que deje de escuchar un canal(*leave*), empiece a escuchar otro(*join*) o responder(*report*) [4]. En una red IPTV, el protocolo mantiene una sesión entre dos dispositivos:

Cliente IGMP Son los responsable de generar los mensajes *leave* o *join* durante la sesión, además de contestar a los menajes *report* que genera el router IGMP. Un ejemplo de este tipo de dispositivos sería nuestro propio PC o un STB.

Router IGMP Reciben los mensaje de los clientes IGMP y envían mensaje periódicos a estos para evitar errores y verificar las peticiones.

A continuación se describen los mensajes IGMP utilizados en las sesiones IPTV.

Membership Query Son enviados por los routers IGMP. Se envían cada cierto tiempo para preguntar a los clientes IGMP a qué árbol *multicast* están suscritos y, en caso de no recibir respuesta, abandonar el árbol.

Membership Report Son enviados por los clientes IGMP. Se utilizan para pedir la suscripción a un árbol *multicast* o para responder a los mensajes *Membership Query* del router IGMP. Si un segundo cliente IGMP observa que otro cliente ha enviado un mensaje de este tipo al mismo grupo *multicast*, el segundo cliente no envía su propio mensaje con el objetivo de ahorrar recursos.

Leave Group Son enviados por los clientes IGMP. Si alguno de los clientes desea abandonar su árbol debe enviar este mensaje al router IGMP.

Por último se describe como sería una sesión IGMP en la que un cliente pide un canal, por ejemplo Antena3 [5], y después abandona el canal.

En primer lugar, el cliente IGMP envía un mensaje *Membership Report* con dirección IP destino al árbol *multicast* de Antena3. Una vez que el router IGMP se haya suscrito al árbol correspondiente, el cliente recibe el contenido multimedia correspondiente. De forma periódica, el router IGMP envía un mensaje *Membership Query* con IP destino a todos los clientes IGMP(224.0.0.1), y este debe contestarle con otro mensaje *Membership Report* con dirección IP destino al árbol *multicast*. Por último, en el momento que el usuario deja de visualizar el canal, el cliente IGMP envía un mensaje *Leave Group* con IP destino a todos los routers IGMP(224.0.0.2) para notificar que este deja el árbol.

En la figura 2.1 se muestra una representación de esta sesión.

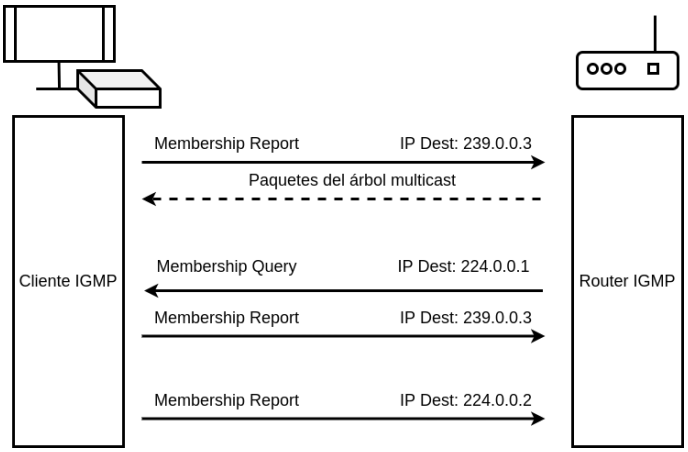


Figura 2.1: Ejemplo de sesión IGMP

2.4. RTP

RTP es un protocolo de la capa de aplicación que proporciona los medios para transmitir contenido multimedia en tiempo real sobre IP utilizando variedad de protocolos en la capa de transporte como User Datagram Protocol (UDP) , Transmission Control Protocol (TCP) y Datagram Congestion Control Protocol (DCCP) [6]. En los servicios IPTV el protocolo RTP utiliza UDP como protocolo de transporte debido a que es menos pesado que TCP y DCCP, y porque es más importante una entrega rápida de los paquetes que la fiabilidad de la misma. Otra de las características de RTP en IPTV es que es compatible para el transporte de datos *multicast*, por lo que se vuelve muy necesario para servicios *Live Television*. Por último, RTP ofrece facilidades para mejorar la calidad de experiencia de los usuarios, como por ejemplo la detección de pérdidas de paquetes o la entrega ordenada de paquetes gracias al número de secuencia que trae en su cabecera [7].

2.5. Calidad de servicio

En términos generales, la calidad de servicio se define como el efecto general del desempeño de un servicio que determine el grado de satisfacción percibido por los usuarios y como los niveles de calidad en el funcionamiento de la red [8]. Al ser este proyecto una herramienta de monitorización, se optó por realizar mediciones en tiempo real de algunos parámetros que conforman la calidad de servicio.

Jitter Se trata de la variación en el tiempo de retardo entre paquetes de un determinado flujo. En nuestro estudio se considera que el *jitter* es inadecuado cuando es mayor de 30ms [9].

Pérdidas de paquetes Número de paquetes que no han sido entregados y/o que han llegado desordenados. Durante la monitorización se considera que el porcentaje de pérdidas es

malo cuando el porcentaje de la media es mayor que $1\,\%$ [10].

Anomalías IGMP Número de errores que se generan en el protocolo IGMP. Dentro de estos errores se encuentran no recibir los paquetes correspondientes a los árboles de difusión suscrito o recibir paquetes de árboles a los que no estamos suscritos.

Throughput Tasa media de bits recibidos por segundo en cada intervalo para cada árbol *multicast*.

Retardo Tiempo que tarda un paquete en llegar a su destino. En este caso, se calcula el retardo total que hubo en cada intervalo para cada árbol de difusión.

Estos parámetros y otros valores calculados son medidos en intervalos de 10 segundos siguiendo la recomendación del ITU-T [11].

2.6. Calidad de experiencia

La calidad de experiencia se trata de la aceptabilidad de una aplicación a través de la subjetividad de los usuarios finales [12]. De igual manera que en el apartado anterior, al ser una herramienta de monitorización se pensó en predecir la calidad que recibe el cliente durante las sesiones mediante la estimación del MOS, una medida ponderada y subjetiva que representa la calidad percibida por los clientes [10]. Se considera que el valor de MOS comienza a ser malo cuando es menor a 4, y crítico cuando es inferior a 3.

Debido a que la estimación del MOS se realiza con datos que se utilizan en la medición de la calidad de servicio, la estimación también se realiza al mismo tiempo que estas mediciones.

2.7. Conclusiones

Este capítulo ha servido para introducir las diferentes tecnologías que se van a utilizar en el proyecto. Se comenzó introduciendo al protocolo IPTV, su diferencia con otros métodos de transferencia multimedia y los tipos de servicios dependiendo del método de difusión. Se continuó describiendo el protocolo IGMP y cómo sus mensajes permite a los host y routers ir cambiando entre los diferentes flujos multimedia. Después, se hizo un resumen del protocolo RTP y se expuso las ventajas que proporciona este protocolo para la distribución de contenido multimedia por *multicast*. Por último se definió la calidad de servicio y la calidad de experiencia, y cuales son medidas y estimadas en el proyecto. En los próximos puntos se muestra el diseño y desarrollo del proyecto y cómo se utiliza esta información.

3

Análisis

3.1. Introducción

Este capítulo se centra en describir la funcionalidad que debe cumplir el proyecto. Además, se incluye una lista con los requisitos funcionales de la herramienta y se describen los requisitos no funcionales.

3.2. Funcionalidad de la aplicación

En este apartado se expone los objetivos y funcionalidades que lleva a cabo la herramienta.

En primer lugar, se debe identificar el tráfico IPTV que este recibiendo los distintos clientes de la red local con la finalidad de controlar a qué árboles se encuentran suscritos cada usuario, y obtener los datos para las mediciones de estos mismos.

A continuación, se debe controlar que los mensajes de unión y abandono a los árboles se hace correctamente, y que los usuarios vayan notificando al router de qué canales están viendo cuando este se lo pida tal como se dicta en el protocolo IGMP. En el caso de que no cumpla el protocolo se considera que se han producido anomalías IGMP. Por otro lado, en los paquetes RTP pertenecientes a los árboles se debe obtener datos como la cantidad de paquetes, los números de secuencia, el retardo entre paquetes sucesivos o los bytes de estos paquetes.

Además, se contabilizan los paquetes UDP que no pertenezcan a ninguno de los árboles de difusión para comprobar si existiera tráfico que afectara a la experiencia del cliente.

Una vez se haya obtenido los datos necesarios, la aplicación debe calcular las medidas de calidad oportunas, estimar la calidad de experiencia del cliente y volcar toda la información obtenida hasta el momento en la base de datos. En esta misma línea se debe incluir la funcionalidad de limpiar y conectarse a la base de datos antes de cada ejecución.

Por último, los datos son representados a través de gráficas que muestra su progresión en cada

intervalo de 10 segundos.

3.3. Requisitos funcionales

Una vez descrito la funcionalidad que debe llevar a cabo nuestra aplicación en la sección anterior, se puede enumerar los requisitos funcionales que debe cumplir la herramienta:

RF-1.– Analizar el tráfico del fichero que se le pase como argumento.

RF-2.– Filtrar el tráfico para analizar una serie de paquetes.

- **RF-2.1.–** Filtrar paquetes IGMP para ver los canales que ve cada usuario.
- **RF-2.2.–** Filtrar paquetes RTP para obtener las mediciones.

RF-3.– Detectar anomalías IGMP.

- **RF-3.1.–** Recibir paquetes de un árbol donde ningún cliente esta suscrito.
- **RF-3.2.–** Estar suscrito a un árbol y no recibir paquetes de este.

RF-4.– Detectar que un árbol tiene algún cliente suscrito.

RF-5.– Diferenciar árboles de difusión con tráfico multimedia de tráfico con parámetros de configuración.

RF-6.– Llevar la cuenta de paquetes de cada árbol de difusión.

RF-7.– Obtener el retardo de paquetes sucesivos de cada árbol de difusión.

RF-8.– Obtener el retardo al cuadrado de paquetes sucesivos de cada árbol de difusión.

RF-9.– Guardar el número de paquetes perdidos de cada árbol de difusión.

RF-10.– Llevar la cuenta de paquetes UDP que no sea de árboles de difusión.

RF-11.– Calcular *jitter*.

RF-12.– Estimar MOS.

RF-13.– Volcar la información actual de la aplicación a la base de datos.

RF-14.– Recuperación ante errores.

- **RF-14.1.–** No parar la ejecución de la aplicación.
- **RF-14.2.–** Continuar con el análisis del tráfico.

RF-15.– Conectar a la base de datos.

RF-16.– Limpiar la base de datos.

RF-17.– Representación de que canal está viendo cada cliente

RF-18.– Representación *jitter*.

RF-19.– Representación MOS.

RF-20.– Representación del porcentaje de pérdidas.

RF-21.– Representación de anomalías IGMP

RF-22.– Representación de cantidad de paquetes capturados.

RF-23.– Representación de tasa media de megabits por segundo.

RF-24.– Representación de tiempo entre llegadas entre paquetes.

RF-25.– Representación de cantidad de paquetes UDP no pertenecientes al estudio.

RF-26.– Representación de medias globales de MOS, *jitter*, porcentaje de pérdidas, anomalías IGMP, *throughput* y

cantidad de paquetes UDP no pertenecientes al estudio.

3.4. Requisitos no funcionales

Este apartado se centrar en listar los requisitos no funcionales que debe cumplir nuestra herramienta y explicar las razones de estas.

Portabilidad La aplicación debe ejecutarse en cualquier equipo con motor Linux y compilador GNU.

Usabilidad La aplicación debe ser sencilla de usar, limitando a los clientes a introducir los parámetros que se necesitan para la ejecución y acceder al navegador para su visualización.

Rendimiento Al ejecutarse en tiempo real, la programación debe ser eficiente para evitar perder datos en el análisis.

Disponibilidad La aplicación debe estar preparada para ejecutarse en cualquier momento.

Confiabilidad Al ejecutarse en tiempo real, hay que evitar su detección en el caso de que surjan errores.

Fiabilidad Los datos que se obtengan del análisis deben de ser correctos respecto al flujo que se estudie.

3.5. Conclusiones

Este capítulo ha descrito la funcionalidad que debe presentar la herramienta y para ello se han presentado tanto los requisitos funcionales como no funcionales que debe cumplir la aplicación. Como resumen, la aplicación debe ser capaz de filtrar el tráfico que se utiliza en el análisis del que no, obtener los datos necesarios de los paquetes que se vayan filtrando, volcar la información a la base de datos y representar estos posteriormente. En el siguiente capítulo se explicará de que forma se realizaran las funcionalidad expuestas.

4

Diseño

4.1. Introducción

Este capítulo se centra en explicar las decisiones que han sido tomadas para el diseño del proyecto como el lenguaje de programación, la base de datos utilizados y la aplicación de representación de datos. Se incluye una división por módulos de la herramienta junto con la descripción de cada uno de estos. Por último, se incluye el diagrama de clases y modelo entidad-relación utilizados.

4.2. Lenguaje de programación

Una de las primeras decisiones que se tomaron fue qué lenguaje de programación usar para el desarrollo de la aplicación. Se propusieron los lenguajes C, Java y Python ya que se han visto durante el grado.

El primer lenguaje que se propuso fue Java porque actualmente se trata de uno de los lenguajes más popular y porque posee la librería Pcap4J [13] para el análisis del tráfico de la red, pero debido a lo poco que se ha practicado con este lenguaje paso a ser descartado rápidamente.

Python fue el siguiente lenguaje que se pensó ya que se tenía más conocimiento de este lenguaje para el análisis del tráfico a diferencia del anterior, librerías como socket [14] para el análisis de paquetes y además se había utilizado para este mismo enfoque en las prácticas de las asignaturas del grado. Aunque Python presenta buenas razones, se descartó debido a las dudas del rendimiento al tratarse de un lenguaje intérprete frente a lenguajes compilados como C++.

Por último, se pensó en utilizar C ya que se trataba de un lenguaje que se había visto desde el principio de la carrera y ya se tenía bastante experiencia en su utilización. Además, al igual que con Python, C ya se había utilizado en prácticas del grado para realizar análisis del tráfico de la red y existen librerías como libpcap [15] para este tipo de trabajos.

Tras pensar los ventajas y desventajas de los lenguajes se decidió utilizar C debido a su rendimiento, librerías que existen para este tipo de trabajo y su experiencia.

La figura 4.1 muestra un resumen de las decisiones tomadas en diagrama Red Amber Green (RAG)
.

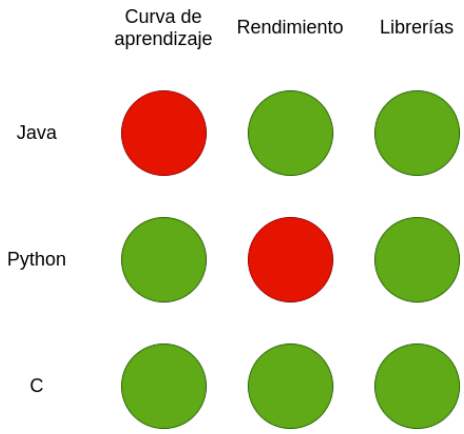


Figura 4.1: Diagrama RAG de lenguajes de programación

4.3. Base de datos

La siguiente decisión a tomar fue que lenguaje de base de datos se iba a utilizar. En un primer momento se propusieron los lenguajes de Elasticsearch y SQL.

Esta decisión fue más fácil de tomar que el tipo de lenguaje de programación porque SQL si se había estudiado a lo largo del grado pero Elasticsearch se trataba de una tecnología totalmente nueva y su curva de aprendizaje sería demasiado alta para este proyecto.

Una vez que se concretó el lenguaje de SQL, lo siguiente fue elegir que gestor utilizar y para ello se presentaron tres posibilidades: MySQL, PostgreSQL y SQLite.

La opción que se tomó en un principio fue SQLite porque se trata de un gestor ligero comparado con las otras dos posibilidades, guarda la información en ficheros de textos y no requiere de ningún tipo de dependencias externas, lo que se convierte en un gestor eficaz para nuestra herramienta. Al final se tuvo que descartar esta opción ya que la aplicación que se eligió para la representación de los datos no era capaz de soportar este formato.

El segundo gestor que se pensó fue PostgreSQL ya que se trataba del gestor que se había utilizado a lo largo del grado y por lo tanto ya se tenía un conocimiento previo de este. Se trata de un gestor de código abierto y, por lo tanto, era soportado por cualquiera de las aplicaciones de representación de datos que se habían propuesto, pero si comparamos su rendimiento con otros gestores como MySQL, PostgreSQL se queda atrás.

El último gestor que se pensó fue MySQL debido a que no era tan ligero como SQLite, ni se tenía un conocimiento previo a diferencia de PostgreSQL. MySQL se trata de un gestor bastante popular por

lo que se tiene bastante documentación y es compatible con la mayoría de aplicaciones que requieren conectarse a una base de datos. Su desarrollo esta enfocado a priorizar la rapidez de las consultas, pero tiene problemas con grandes volumenes de datos a diferencia de PostgreSQL y la complejidad de las consultas están muy limitadas.

Tras deliberar las ventajes y desventajas de las opciones [16], se eligió MySQL debido a su rapidez en las consultas ya que las consultas que se iban a realizar no serían muy complejas. En el caso de que se necesitara más volúmenes de datos, la opción elegida habría sido PostgreSQL.

La figura 4.2 muestra un resumen de las decisiones tomadas.

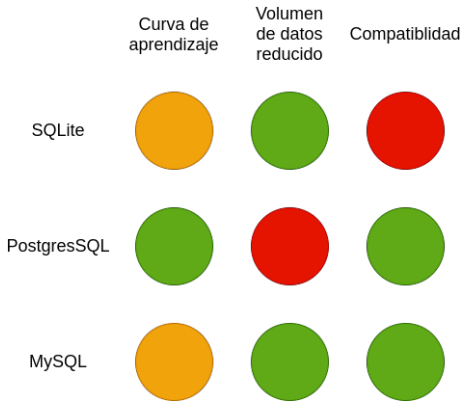


Figura 4.2: Diagrama RAG de bases de datos

4.4. Aplicación de representación

Como parte de la funcionalidad de la herramienta los datos que se vayan recopilando en la base de datos deben ser representados posteriormente para monitorizar el flujo de la sesión de los clientes. Una vez que se decidió cómo se iban a guardar los datos, el siguiente paso fue decidir cómo se iban a representar estos. La forma más cómoda de visualizar la progresión de los datos para los usuarios sería a través de gráficas porque de esa manera se puede observar los picos que aparecerían en las distintas medidas o la falta de algún tipo de dato, y para ello se pensó en utilizar un software dedicado a esta tarea. Después de investigar entre distintas aplicaciones se pensó dos posibilidades: Kibana [17] y Grafana [18].

Esta vez la decisión fue rápida ya que Kibana no es compatible con bases de datos que fuera distinta a Elasticsearch. Por lo tanto, dado que se tenía cierta conocimiento de Grafana, esta fue la opción elegida.

La figura 4.3 muestra un resumen de las decisiones tomadas.

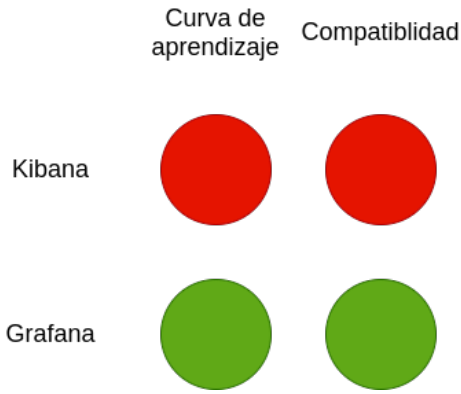


Figura 4.3: Diagrama RAG de aplicación de representación

4.5. Diseño de la herramienta

En esta sección se describe cómo se han implementado los medios necesarios para cumplir las funcionalidades descritas en el capítulo anterior. La aplicación se encuentra dividida en módulos con el objetivo de agrupar aquellos requisitos que tengan cierta dependencia de otros. La figura 4.4 muestra la división por módulos.

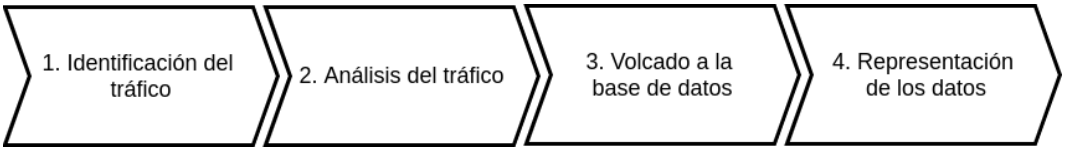


Figura 4.4: Módulos de la aplicación

4.5.1. Módulo de identificación del tráfico

Este módulo es el encargado de identificar el tráfico de una sesión IPTV que se encuentra en la red local para realizar los análisis posteriores.

En primer lugar, el tráfico que recibe nuestra herramienta se encuentra en un fichero de red(**pcap**) que se obtienen mediante **TCPdump** [15] usando las líneas de comando correspondientes o con Wireshark [19].

Al tratarse de un tráfico de una sesión IPTV, los paquetes correspondientes a la sesión se transmiten por UDP, pero para averiguar que paquetes UDP pertenecen a una sesión se necesita comprobar los paquetes IGMP que envía el usuario para saber a que árbol de difusión se encuentra suscrito. La manera de comprobar el tipo de paquete será buscando en la cabecera Ethernet de cada paquete.

En resumen, se filtran los paquetes IGMP y UDP del tráfico que reciba nuestra herramienta para obtener los paquetes correspondientes a una sesión IPTV y realizar un análisis a estos paquetes.

4.5.2. Módulo de análisis del tráfico

El siguiente módulo se encarga del análisis de los paquetes IGMP y RTP.

En el caso de los paquetes IGMP se comprueba si los mensajes de unión o abandono a los árboles se hace correctamente, y que los usuarios vayan notificando al router de qué canales están viendo como se dicta en el protocolo. En el caso de que no cumpla el protocolo se considera que se han producido anomalías IGMP. La comprobación del tipo de mensaje se realiza buscando en la cabecera IGMP del paquete y la respuesta de los usuarios ante las preguntas del router se controla con los tiempo de recepción y envio de los respectivos paquetes.

Por otro lado, en los paquetes UDP se debe obtener los datos correspondientes al análisis como cantidad de paquetes recibidos del árbol, el retardo entre paquetes sucesivos o los bytes de estos paquetes. Con el objetivo de conseguir ciertos datos se deberá acceder a la cabecera RTP de los paquetes UDP.

Por último, se obtiene el número de paquetes y tiempo de recepción de los paquetes UDP que no pertenezcan a ninguno de los árboles de difusión.

4.5.3. Módulo de volcado a la base de datos

El módulo de volcado a la base de datos se basa en calcular algunas medidas de calidad como el *jitter* o estimar la calidad de experiencia de MOS, y guardar la información obtenida hasta el momento en la base de datos.

Después de haber guardado los datos obtenidos en el módulo anterior, estos son guardados en las estructuras de datos correspondientes para ser volcados a la base de datos al concluir el intervalo de tiempo.

Una vez concluido el intervalo de tiempo, la herramienta procede a calcular las calidades de servicios necesarias y a estimar el MOS con los datos del intervalo actual y del anterior. Una vez volcado los datos a la base de datos la herramienta continua analizando el tráfico.

Por último, en este módulo también reside la funcionalidad de limpiar y conectarse a la base de datos antes de cada ejecución.

4.5.4. Módulo de representación de los datos

El último módulo consiste en la representación de los datos guardados en la base de datos a través de la aplicación de monitorización Grafana.

Los datos son representados en gráficas que muestra su progresión en intervalo de 10 segundos.

Además, se incluyen paneles con la media, de todas las sesiones IPTV, de algunas calidades de servicio y del MOS. Las gráficas que se muestran son las mismas que se enumeraron en los requisitos funcionales.

4.6. Diagrama de clases

La siguiente sección muestra el diagrama de clases utilizado en el proyecto. Al no utilizar un lenguaje de programación orientado a objetos se ha adaptado el diagrama para mostrar los TADs desarrollados. La figura 4.5 muestra el diagrama de clases implementado.

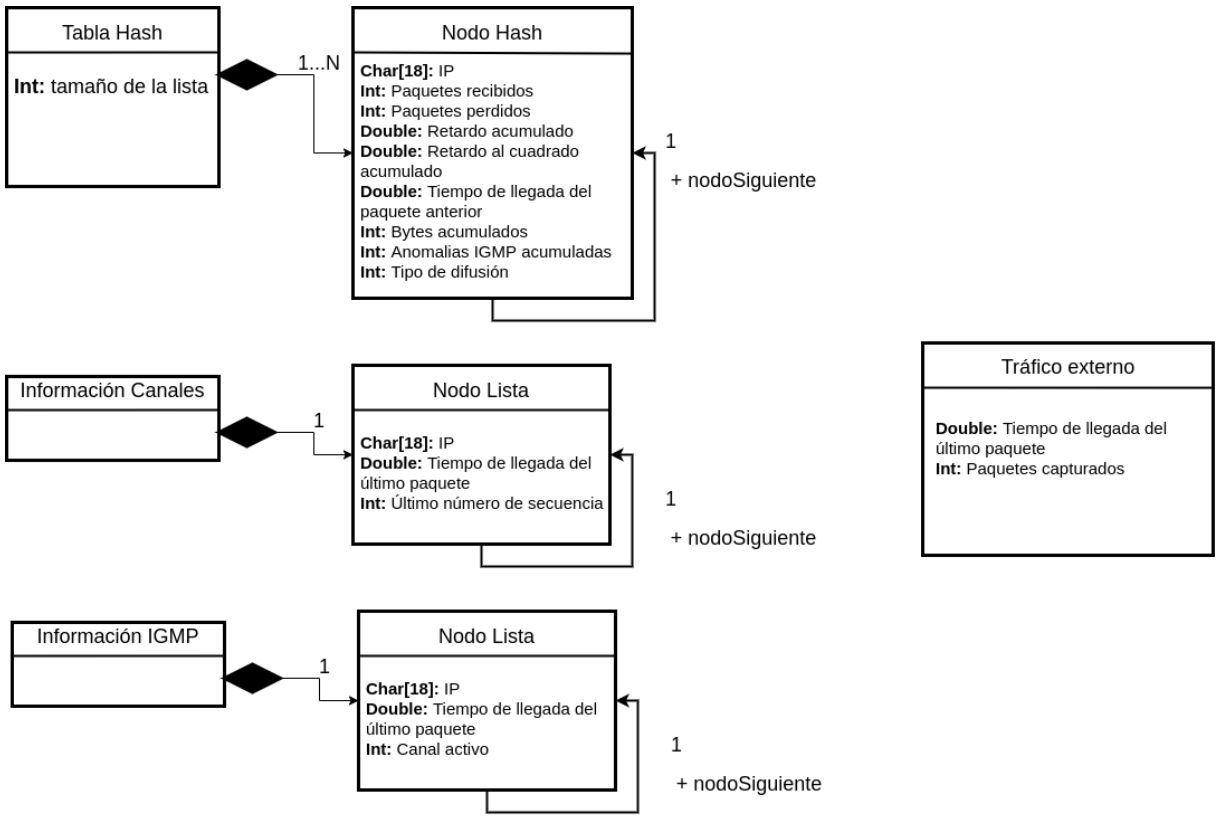


Figura 4.5: Diagrama de clases

La tabla hash se encarga de guardar la información de las sesiones IPTV que se volcará posteriormente a la base de datos. Los TAD de información de Canales e IGMP se encargan de guardar datos que usan en el calculo de los anteriores. Por último, el TAD de tráfico externo es información que también se volcará a la base de datos, pero que no se incluye en la tabla hash por no pertenecer a sesiones IPTV.

4.7. Modelo entidad-relación

La figura 4.6 muestra el modelo entidad-relación implementado. Las entidades son independientes entre ellas debido a que la información que guarda cada una de ellas se utilizará en gráficas diferentes.

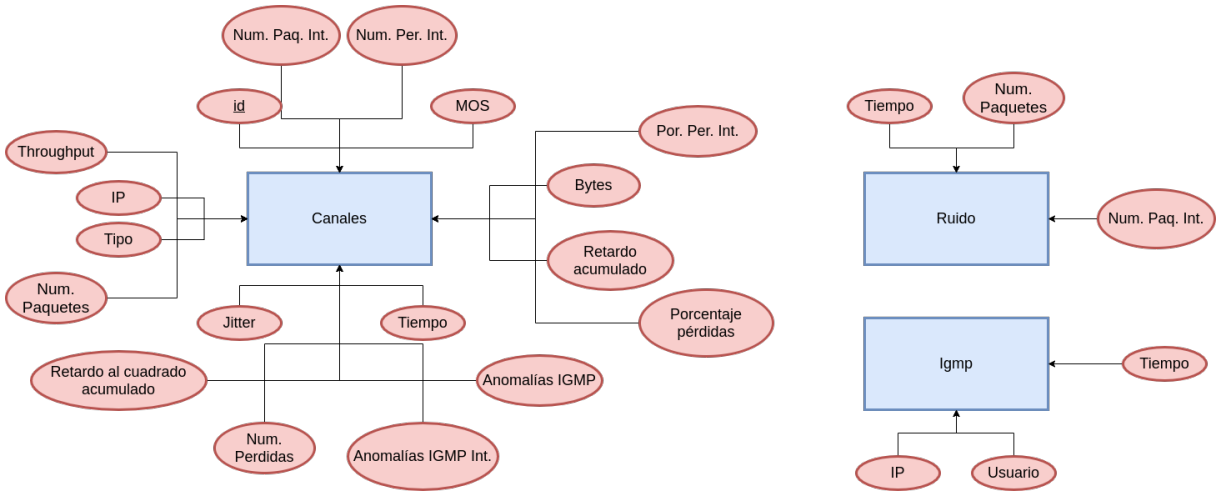


Figura 4.6: Modelo Entidad-Relación

La entidad Canales guarda las calidades de servicio a representar al igual que la calidad de experiencia estimada, la entidad Igmp se utiliza para mostrar los canales que están viendo cada uno de los clientes y la entidad Ruido guarda los datos referente al tráfico UDP externo que no tienen que ver con las sesiones UDP. La razón de que la base de datos no se encuentre normalizada es debido que puede afectar al rendimiento de las consultas a la hora de representar los datos.

4.8. Conclusiones

Este capítulo ha explicado las decisiones que se han tomado para el desarrollo de la herramienta. En primer lugar, el lenguaje de programación fue C por disponer de librerías compatibles para nuestro proyecto, un rendimiento óptimo y por tener experiencia en el lenguaje. En segundo lugar, como base de datos se ha utilizado MySQL por ser compatible con la aplicación de representación de datos y por alto rendimiento. Por último, la aplicación de representación elegida fue Grafana por su tener experiencia en su uso.

Además, se ha introducido una separación por módulos de la funcionalidad de la herramienta para facilitar un desarrollo por partes y se ha mostrado el diagrama de clases y el modelo de entidad-relación que se usarán en el desarrollo.

5

Desarrollo

5.1. Introducción

El capítulo actual explica de forma detallada y técnica el desarrollo de la aplicación con el objetivo de extender lo que se vio en el capítulo anterior. La figura 5.1 muestra un esquema del funcionamiento de la herramienta, siendo las partes escritas en rojo las que han sido desarrolladas en este trabajo. Por último, se describe las estructuras de datos utilizadas en el proyecto y la conexión que hay entre ellas.

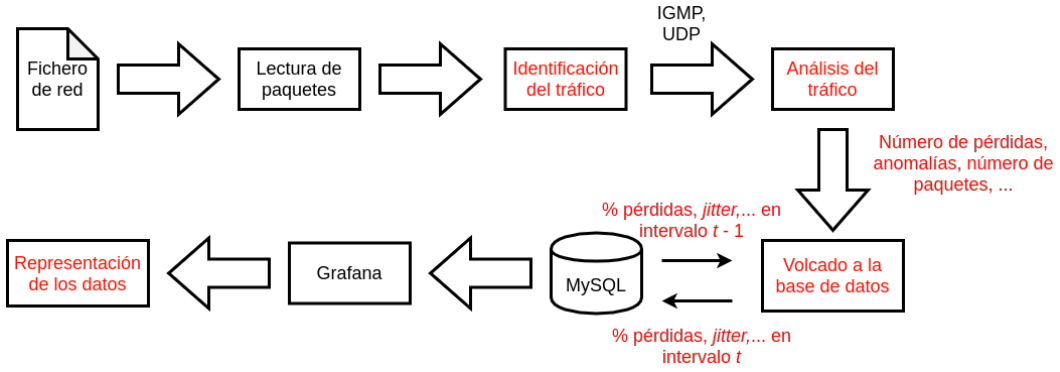


Figura 5.1: Esquema de desarrollo

5.2. Identificación del tráfico

La primera parte de la aplicación consiste en la lectura del fichero que contiene el tráfico, e identificar los paquetes IGMP y UDP que pertenecen a una sesión IPTV. La forma en que se leen los paquetes y se obtienen campos de estos ha sido posible a las funciones integradas en la librería **libpcap**.

En primer lugar, el usuario debe introducir como parámetro de entrada el fichero que se desea analizar y un flag para indicar que se desea aplicar la función de monitorización.

En el momento de leer el primer paquete se guarda el valor del campo *epoch time*, de esta forma se

tiene una referencia de tiempo para después volcar la información que se vaya obteniendo o analizar las anomalías IGMP. Después, se accede al campo *type* de la cabecera Ethernet y se compara con **ETHERTYPE_IP** para comprobar si es un paquete IP. Si no se trata de un paquete IP este se descarta porque no puede ser ni UDP ni IGMP.

Por último, una vez que sabemos que el paquete es de tipo IP, se accede a esta cabecera y se obtiene el campo *protocol* con el cual se hace una comparación con los valores **IPPROTO_IGMP** e **IPPROTO_UDP** para saber si se trata de un paquete IGMP o UDP respectivamente. En el caso de que no sea igual a ninguno de los anteriores se descarta el paquete.

El único problema que se tuvo a la hora de desarrollar este módulo fue recordar como moverse entre los distintos campos de las cabeceras y obtener los valores de estos.

Una vez concluido el módulo, la aplicación dispone de los paquetes que deben analizarse en el siguiente módulo.

5.3. Análisis del tráfico

El siguiente módulo se encarga del análisis de los paquetes filtrados en la anterior sesión. En concreto, para los paquetes IGMP se procede a identificar los árboles de difusión *multicast*, qué clientes lo han solicitado y vigilar las posibles anomalías que podrían surgir de este protocolo mientras que, para los paquetes UDP, se procede a identificar aquellos que tienen encapsulado la capa RTP y obtener datos de estos.

En el caso de que llegue un paquete IGMP el programa se sitúa en la cabecera de este y comprueba que tipo de mensaje es. Si se trata de un mensaje *Membership Report* pueden ocurrir dos casos:

Grupo registrado El cliente está solicitando unirse al grupo *multicast* existente o está contestando a un mensaje *Membership Query* y por lo tanto hay que guardar la dirección IP del cliente para ese grupo en la aplicación, registrarlo en el grupo correspondiente, guardar el tiempo de recepción y marcar al grupo de que esta siendo usando.

Grupo no registrado Se trata del primer cliente que solicita ese grupo *multicast* y por lo tanto se debe reservar espacio para ese árbol y se registrar al cliente posteriormente.

Por otro lado, si se trata de un mensaje *Leave Group* se elimina la IP del cliente del grupo correspondiente. En el caso de que el grupo se quede vacío marcamos al grupo de que no esta siendo usado.

Si se trata otro tipo de mensajes se ignora ya que no entra dentro de nuestro estudio.

En la figura 5.2 se muestra un diagrama de los pasos anteriores.

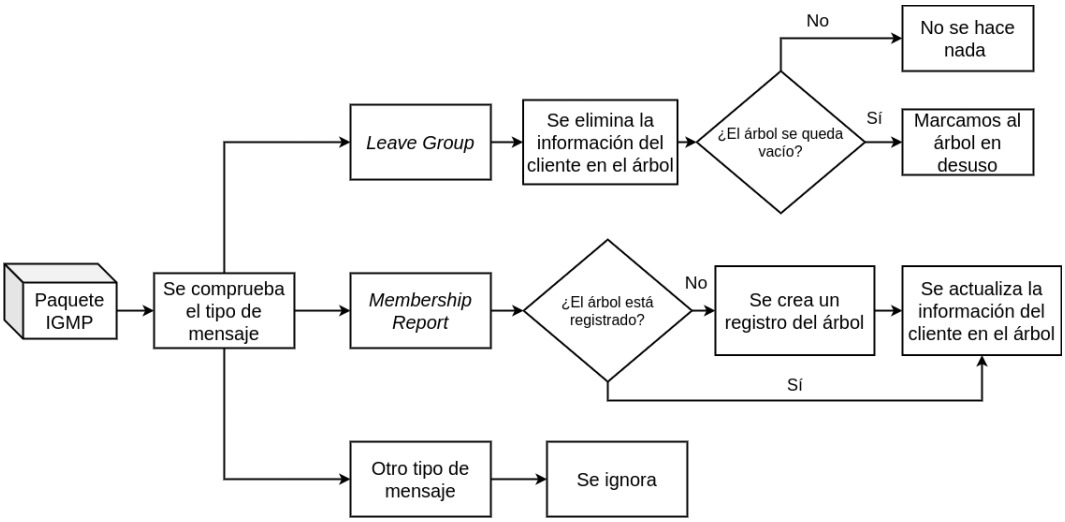


Figura 5.2: Esquema de procesamiento de paquetes IGMP

Debido a que IPTV utiliza IGMP en versión 2 nos quitamos la preocupación de tener que tratar con mensajes de IGMP de otra versión distinta.

En el caso de ser paquetes UDP comprobamos si la dirección IP destino corresponde con alguna de las direcciones de los árboles *multicast* registrados, y de no ser así se contabiliza como paquete UDP de otro tráfico. En el caso de corresponder con alguna de las direcciones se accede a la cabecera RTP y se comprueba la versión RTP que se utiliza para averiguar si el árbol transmite contenido multimedia, si se trata de tipo 2, o parámetros de configuración. A continuación, tras pasar los filtros anteriores, se actualizan los datos del grupo pertenecientes al paquete aumentando el número de paquetes recibidos de ese grupo, tomando como el último tiempo de recepción el de este, y se guarda la diferencia de tiempo de recepción de este paquetes con el anterior y los bytes del paquete. Por último, se guarda el número de secuencia del paquete y se comprueba si ha habido pérdidas utilizando el número de secuencia del anterior paquete recibido del mismo árbol.

Uno de los problemas que se tuvo a la hora de desarrollar esta parte fue de cómo se iba a diferenciar los paquetes RTP que enviaban contenido multimedia de los que enviaban valores de configuración, y la solución fue comprobar el valor del capo tipo en la cabecera RTP como se describió anteriormente.

En la figura 5.3 se muestra un diagrama de los pasos anteriores.

La detección de anomalías IGMP se obtiene en dos situaciones. La primera, tras comprobar la versión RTP, se comprueba si el paquete pertenece a un grupo que este siendo usando por algún cliente, y de no ser así se contabiliza como anomalía IGMP. El segundo caso se realiza cada vez que trascurre un segundo en el flujo, se comprueba si el tiempo de llegada del último paquete UDP, de un árbol activo, es menor que el último paquete IGMP enviado por alguno de los clientes, y de ser así se contabiliza como anomalía IGMP. El primer caso comprueba si existe algún problema con el envío de mensajes IGMP por parte de los clientes, mientras que el segundo caso comprueba que los paquetes

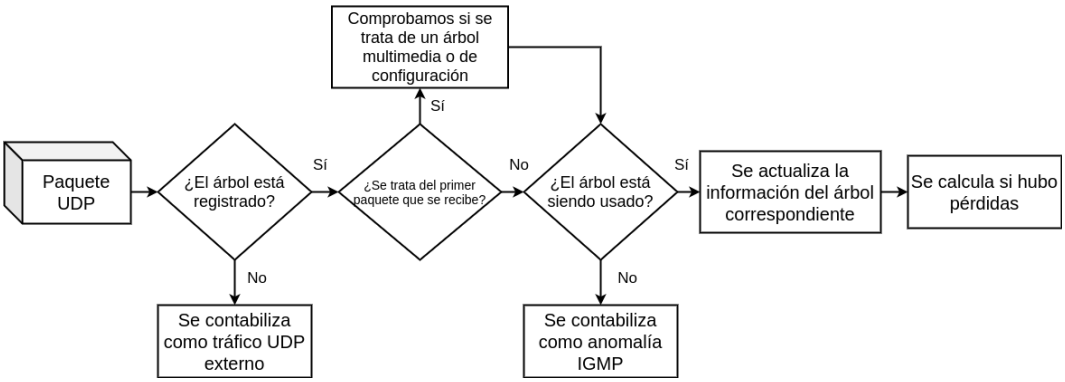


Figura 5.3: Esquema de procesamiento de paquetes UDP

que se transmiten por los árboles de difusión *multicast* no se realiza correctamente.

5.4. Volcado a la base de datos

El módulo comienza a ejecutarse una vez que se haya capturado 10 segundos de flujo y se basa en calcular las calidades de servicio y estimar la calidad de experiencia para después volcar estos y otros datos en la base de datos.

La base de datos esta formada por dos tipos de datos: los datos que se van obteniendo en cada intervalo y los datos que se utilizan en la monitorización. Los datos obtenidos en cada intervalo son valores acumulativos y con ellos se obtienen datos globales del flujo, por ejemplo el porcentaje global de pérdidas. Los datos que se utilizan en la monitorización se obtienen con la diferencia de los datos de intervalos consecutivos y estos no se utilizan en las operaciones para calcular otros datos.

En primer lugar se debe obtener los siguientes valores de la base de datos para cada grupo:

- Sumatorio de la diferencia de tiempo de llegada entre paquetes.
- Sumatorio de la diferencia de tiempo de llegada al cuadrado entre paquetes.
- Número de paquetes recibidos.
- Número de paquetes perdidos.
- Número de anomalías IGMP.
- Número de bytes recibidos.

Una vez que se tiene los valores del intervalo anterior, el siguiente paso es calcular la diferencia entre el intervalo actual y el intervalo anterior para estos datos, ya que la diferencia será el valor que se representará en la herramienta de visualización como se ha dicho anteriormente.

Posteriormente, se calcula el valor de jitter, retardo del intervalo, la tasa de megabits por segundo del intervalo, porcentaje de pérdidas totales, porcentaje de pérdidas del intervalo y la estimación del MOS del intervalo para cada grupo de difusión para insertarlos en la base de datos como se muestra

en la figura 4.6.

Una vez volcado los datos anteriores falta por guardar información relacionada con IGMP y el tráfico externo UDP. Con respecto a la información IGMP, se guarda en la base de datos el tiempo de recepción del último paquete recibido de cada grupo junto con la dirección IP del cliente que este suscrito al grupo y la dirección IP del grupo de difusión. De forma similar, la información que se vuelca en la base de datos del tráfico UDP externo es el tiempo de recepción del último paquete de este tráfico, la cantidad de paquetes recibidos y la cantidad total de paquetes recibidos.

El problema que se encontró en este módulo fue en que momento se tenía que realizar el volcado de los datos. Tras investigar, se encontró un documento oficial de la ITU que aconsejaba que las medidas y estimaciones para este tipo de análisis se debía realizar en intervalos de 10 segundos.

Una vez termine este módulo se dispondrá de la información necesaria para representar los datos en el siguiente módulo.

5.5. Representación de los datos

El último módulo consiste en representar los gatos guardados en el módulo anterior mediante gráficas utilizando el software de código abierto Grafana. Se dispone de varios tipos de despliegue de la aplicación, pero para nuestro proyecto se ha decidido instalar Grafana en local.

En primer lugar, Grafana pide que se seleccione el tipo de basde de datos que se va a utilizar y en nuestro caso fue MySQL. Tras seleccionar el tipo de base de datos se debe introducir la dirección a esta y un usuario que tenga permisos para acceder y leer los datos de las tablas.

Una vez que Grafana tenga acceso a la base de datos, se concede la posibilidad de crear los paneles para la representación de los datos seleccionando el tipo de visualización(gráficas, paneles con valores, etc.) y la sentencia SQL para seleccionar los datos que se quieran mostrar. Los paneles que se han utilizado para este proyecto son los mismos que se enumeraron como requisitos funcionales en la sección 3.3, y en cada uno de ellos se representan los valores correspondientes a cada grupo de difusión del mismo tipo(multimedia o parámetros de configuración). Si se desea realizar algún cambio en algunos de los paneles, como cambiar el tipo de los grupos de difusión a mostrar, el cliente puede acceder a los ajustes del panel y efectuar los cambios que le parezcan convenientes.

En este último módulo solo hubo problemas a la hora de crear la gráfica que muestra que canal esta viendo cada cliente. Se tuvo que buscar un pluggin en la tienda de Grafana para utilizar este tipo de gráficas y guardar el tiempo en que los clientes enviaban los mensajes de unión o de respuesta IGMP para llevar un control de los canales que estaban viendo en cada momento.

5.6. Conclusiones

Este capítulo ha servido para explicar como está hecha la aplicación desde una perspectiva más detallada y técnica. Tal como sucedió en el capítulo anterior, se ha descrito la funcionalidad de la aplicación a través de la división por módulos. Además, se ha descrito los datos que acaba mostrando la aplicación y como será su representación. El siguiente capítulo hablará de la validación de la aplicación probando esta ante distintos casos de prueba.

6

Pruebas y resultados

6.1. Introducción

El siguiente capítulo trata sobre las pruebas que se han realizado para comprobar el correcto funcionamiento de nuestra herramienta y como refleja el comportamiento de la red. En primer lugar, se describe como se han conseguido y en que condiciones se han generado los ficheros para los casos de prueba. Por último, se mostrarán los resultados obtenidos por nuestra aplicación.

6.2. Generación de los ficheros de prueba

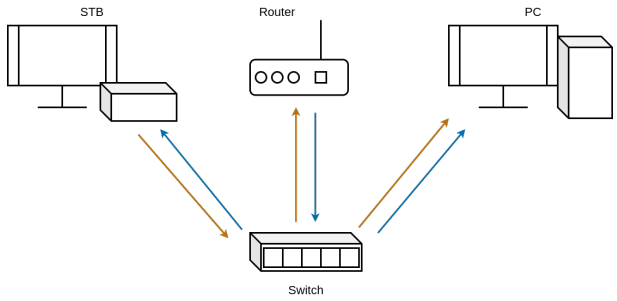
Se utilizaron dos formas para conectarse a un tráfico de difusión *multicast*: utilizando un STB para que posteriormente se visualizara el contenido en la televisión, y usando un PC que se conecta a los grupos y que al mismo tiempo hiciera de reproductor. En ambos casos se utiliza el programa Wireshark para capturar el tráfico proveniente de los grupos de difusión y así guardar los paquetes en los ficheros que se utilizan posteriormente como entrada en nuestro programa.

Captura en *Set-Top-Box* Mientras que el cliente pedía los canales que quería ver a través del STB, el tráfico, tanto entrante como saliente, era duplicado y redirigido al PC mediante un *switch* que se conectaba entre el router y el STB.

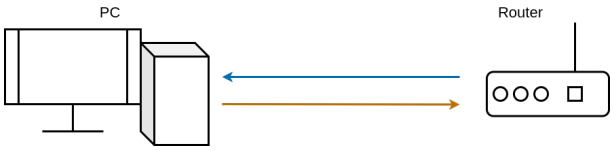
Captura en PC A diferencia que en el método anterior, el PC es el encargado de pedir el tráfico y visualizarlo, y al mismo tiempo ir capturándolo. En ciertos casos se han utilizado dos PC para recibir distinto tráfico y un switch para clonar el tráfico de uno de los PC al PC que realiza la captura. En este caso, se pierde cierto tráfico perteneciente a IPTV, como por ejemplo la guía de programación electrónica.

La figura 6.1 muestra los escenarios de prueba realizados.

Además de la generación de los ficheros, se realizaron ciertas modificaciones a los ficheros capturados o se introdujo tráfico adicional a red mientras que se realizaba la captura para simular ciertas condiciones en la red.



(a) Captura con STB



(b) Captura con PC

Figura 6.1: La figura 6.1(a) muestra como el *switch* redirigía el tráfico del STB hasta el PC y la figura 6.1(b) muestra la captura en el PC mientras que este pedía igualmente el tráfico

6.3. Pruebas de validación

El siguiente apartado describe las pruebas y los resultados obtenidos en los ficheros junto con una descripción de las modificaciones realizadas a los ficheros de prueba o anomalías introducidas a la red mientras se realizaba la captura. Se han dividido las pruebas en subapartados para facilitar la explicación de estas.

6.3.1. Sesión IPTV normal

La siguiente prueba muestra los resultados obtenidos en una sesión IPTV normal, donde el cliente envía los mensajes IGMP correspondientes para unirse al grupo de difusión *multicast* y comienza a recibir los paquetes multimedia del grupo. La captura de esta sesión se relizó capturando el tráfico en el *Set-Top-Box*.

A continuación, la figura 6.2 muestra las gráficas que genera la herramienta al terminar el análisis. La primera gráfica muestra el canal que esta viendo el cliente en cada momento, en este caso el cliente ve únicamente el canal TNT HD. Los seis subpaneles que se encuentran a la derecha muestra medias generales de la sesión, y los que se encuentran en verde actúan de paneles semáforos siendo verde el color para indicar que los valores son de buena calidad, naranja para indicar que una calidad mínima y rojo para indicar que los valores son pésimos. La tabla de MOS se encuentra en 5 ya que al no haber pérdidas el valor se mantiene óptimo y la gráfica de porcentaje de pérdidas se mantiene a 0. Con respecto a la gráfica del *jitter* su valor se mantiene por debajo de 30ms, por lo que su calidad es

buena, y esta no presenta picos, lo que indica que no hubo problemas en el tráfico de la red local. Por último, la gráfica de anomalías IGMP se mantiene en 0 lo que muestra que el intercambio entre el host IGMP y el router IGMP se hizo tal como dicta el protocolo. La figura 6.3 muestra como la cantidad de paquetes, tasa de Mbits/s y el tiempo entre llegada de paquetes mantienen valores estables por la falta de pérdidas de paquetes. Por último, en la misma figura, se encuentra la gráfica del tráfico UDP externo al análisis y se observa un pico que pertenece al tráfico del árbol de difusión antes de ser registrado.

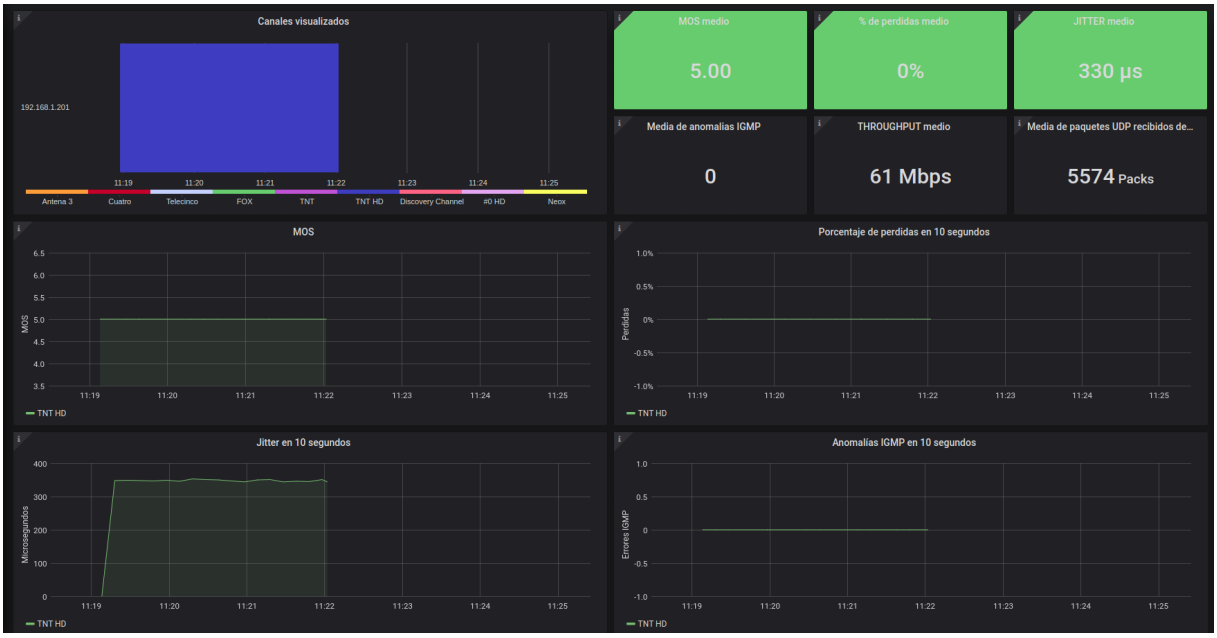


Figura 6.2: Gráficas de calidad de servicio generadas de una sesión IPTV normal

Figura 6.3: Gráficas de información adicional generadas de una sesión IPTV normal

6.3.2. Sesión IPTV con dos clientes

La prueba analiza una sesión IPTV normal con dos clientes en la misma red local, ambos en dos ordenadores. En esta prueba se comprueba como la aplicación reacciona ante el cambios de grupos de difusión *multicast* por parte de los clientes. Como en el caso anterior, no se han provocado pérdidas ni anomalías de ningún tipo y la captura del fichero se ha realizados desde uno de los ordenadores.

La figura 6.4 muestra las gráficas como resultado de este análisis. Al tener dos clientes la primera gráfica muestra que canal esta viendo cada cliente y como estos cambian de canal o dejan de ver la televisión y vuelven a conectarse pasado un tiempo. Al igual que en la prueba anterior, los valores de los paneles con medias del tráfico general indican que la sesión esta siendo correcta, las gráfica de MOS y porcentaje de pérdidas nos indica que ningún canal esta sufriendo pérdidas y que no existen anomalías IGMP. Con respecto al panel del *jitter*, al haber dos clientes usando una sesión IPTV y al realizar las pruebas en dos PC, el *jitter* es más alto que en el caso anterior pero este sigue bajo como para que los clientes tengan una calidad de experiencia muy buena. La figura 6.5 muestra como la cantidad de paquetes recibidos, la tasa de Mbits/s y el tiempo de llegada entre paquetes son estables al no haber pérdidas. Si comparamos estos resultados con los de la anterior prueba se puede observar que los valores entre ambas pruebas no coinciden debido a que los canales en HD requieren de más recursos que los canales con menor resolución [20]. Por último, la gráfica de tráfico UDP externo muestra más picos que en el caso anterior ya que la prueba fue capturada desde un PC y por lo tanto recibe más paquetes de otras sesiones que si se hiciera directamente desde el *Set-Top-Box*.

Figura 6.4: Gráficas de calidad de servicio generadas de una sesión IPTV normal con dos clientes

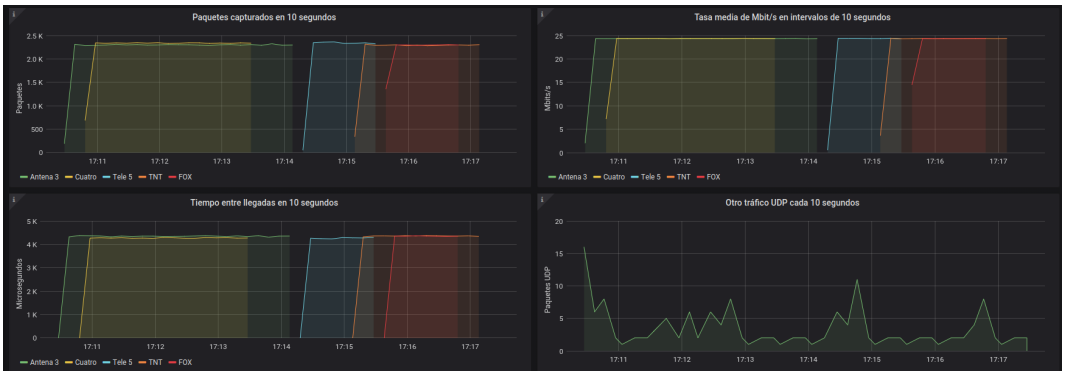


Figura 6.5: Gráficas de información adicional generadas de una sesión IPTV normal con dos clientes

6.3.3. Sesión IPTV con tráfico TCP

La siguiente prueba muestra los resultados obtenidos en una sesión IPTV donde la red local se encuentra lleno de paquetes TCP proveniente de otros tráficos. El objetivo de esta prueba era determinar si el tráfico TCP influía en algún aspecto a la sesión IPTV. La captura del tráfico se realizó desde un PC que pedía el tráfico *multicast* mientras visualizaba varios vídeos de Youtube para obtener el tráfico TCP y capturaba todo al mismo tiempo.

La figura 6.6 muestra las gráficas generadas después del análisis. A pesar del tráfico TCP, los valores medios de la sesión siguen siendo buenos por lo que la experiencia del usuario no fue molesta. Se puede observar que la gráfica de MOS presenta un pico descendente de 0.03 aunque en la gráfica de porcentaje de pérdidas no se observan cambios, esto se debe a que las pérdidas han sido tan pequeñas que no ha influido en la experiencia del cliente. Si se comprueba el tráfico desde Wireshark, se avisa que solo se ha producido una pérdida de un paquete en toda la sesión. Con respecto a la gráfica del *jitter*, los resultados son similar a la prueba con dos clientes porque el ordenador estaba pidiendo varios tráficos al mismo tiempo, lo que se refleja como una pequeña variación del *jitter*. Por último, la gráfica de anomalías IGMP indica que no hubo problemas con el protocolo. La figura 6.7 muestra que los paquetes recibidos, tasa de Mbits/s y tiempo de llegada entre paquetes en los intervalos presenta valores corrientes a pesar del tráfico TCP.

Figura 6.6: Gráficas de calidad de servicio generadas de una sesión IPTV con con tráfico TCP

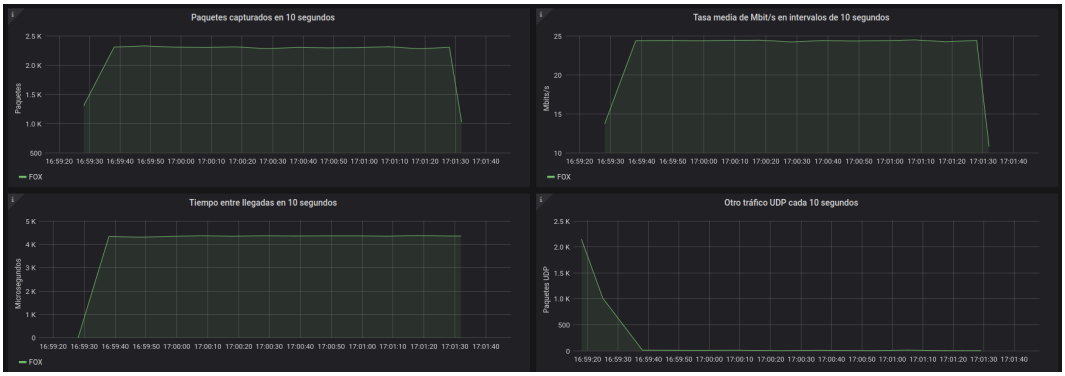


Figura 6.7: Gráficas de información adicional generadas de una sesión IPTV con con tráfico TCP

6.3.4. Sesión IPTV con tráfico UDP externo

La siguiente prueba es similar a la anterior pero en esta ocasión se introduce tráfico UDP que no pertenece a ningún árbol. La forma en se introdujo el tráfico UDP externo fue mediante paquetes enviados por *broadcast* desde un PC en la misma red local, con el objetivo de inundar la red determinar si este influye en algo a la sesión IPTV. La captura del tráfico se realizó desde un PC que se subscribía al árbol de difusión mientras capturaba el tráfico entrante.

La figura 6.8 muestra los resultados del análisis. Se observa que los subpaneles de medias de la sesión nos indica que hay problemas con el MOS y el porcentaje de pérdidas ya que se encuentran en rojo, y esto se puede verificar en las respectivas gráficas. Por otro lado, el *jitter* sigue manteniendo un valor adecuado para la sesión. Con respecto a las gráficas de MOS, porcentaje de pérdidas y *jitter*, las gráficas tienen un comportamiento similar de forma que las tres tienen valores que se encuentran por

encima de lo habitual(figura 6.2) hasta el final donde adquieren valores más corrientes. Esto coincide con la captura que se realizó, donde se comenzó a capturar el tráfico ya inundado por paquetes UDP, y se dejó de inundar antes del final de la captura. Por último, observando la gráfica de anomalías IGMP, el tráfico UDP externo no ha afectado al intercambio de mensajes del protocolo IGMP. En la figura 6.9 se puede observar que el tráfico UDP inyectado provoca que se capturen menos paquetes por intervalo, la tasa de Mbits/s baje y el tiempo de llegada entre paquetes aumente. La gráfica de tráfico UDP externo muestra la cantidad de paquetes inyectados en la red se han recibido por intervalo.

Figura 6.8: Gráficas de calidad de servicio generadas de una sesión IPTV con tráfico UDP

Figura 6.9: Gráficas de información adicional generadas de una sesión IPTV con tráfico UDP

6.3.5. Sesión IPTV con anomalías IGMP

La prueba analiza una sesión IPTV con anomalías IGMP en la sesión, concretamente las anomalías generadas al no recibir tráfico del grupo de difusión al que el cliente está suscrito. La traza de red utilizada para esta prueba se generó capturando una sesión normal de IPTV para posteriormente descartar los paquetes del grupo en una franja de tiempo determinada.

En la figura 6.10 se observan los resultados del análisis. La gráfica de anomalías IGMP muestra como la aplicación detecta la falta de paquetes multimedia a pesar de que los clientes envían los mensajes *Membership Report* para ir contabilizando los errores. La herramienta también detecta las pérdidas ocasionando una bajada del MOS o la subida del *jitter*. La razón de que el MOS de buenos resultados mientras que las pérdidas son altas en los subpaneles es debido a que se tratan de medias globales. La figura 6.11 muestra las pérdidas de paquetes que se provoca en uno de los intervalos al igual que la bajada de la tasa de Mbits/s y de la subida del tiempo entre llegada de los paquetes.

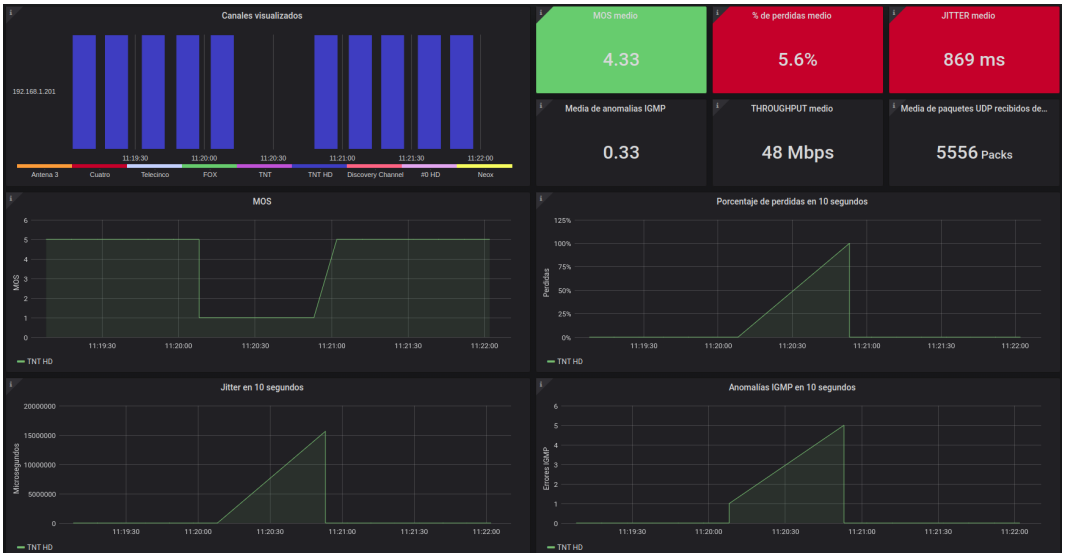
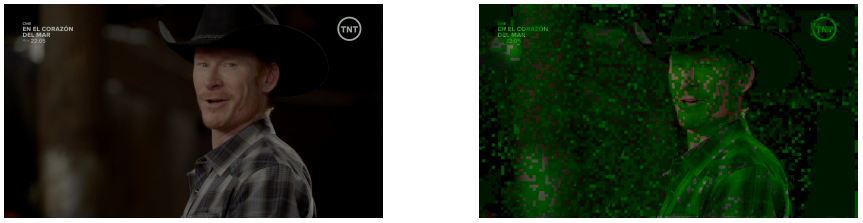


Figura 6.10: Gráficas de calidad de servicio generadas de una sesión IPTV con anomalías IGMP

Figura 6.11: Gráficas de información adicional generadas de una sesión IPTV con anomalías IGMP

6.3.6. Sesión IPTV con pérdidas

La última prueba se centra en comprobar como la aplicación detecta las pérdidas de los paquetes del grupo de difusión. La prueba se realizó con el fichero de red de la prueba 6.3.1 al que se descarto un 1 % de los paquetes multimedia. En la figura 6.12 se muestra una comparativa de un mismo fotograma para demostrar el impacto de las pérdidas.



(a) Fotograma sin pérdidas

(b) Fotograma con 1 % de pérdidas

Figura 6.12: La figura muestra el impacto que sufre un fotograma al producirse pérdidas. La figura 6.12(a) muestra el fotograma real sin pérdidas mientras que la figura 6.12(b) muestra el mismo fotograma con una pérdida uniforma del 1 %

La figura 6.13 muestra los resultados del análisis. En primer lugar, los subpaneles informan de que el valor medio del MOS es malo, y la media de porcentajes de pérdidas coincide con el porcentaje que se generó en el fichero, lo que demuestra que la aplicación calcula en tiempo real las pérdidas en la sesión correctamente. Con respecto a los paneles de las gráficas, la gráfica de MOS contiene picos indicando que en ciertos intervalos ha habido más perdidas que en otros pero acaba oscilando sobre 2.2, la gráfica de porcentaje de pérdidas indica que las pérdidas se mantienen constante en un 1 % de modo que coincide en la forma en que se generaron las pérdidas en el fichero. Por último, la gráfica de *jitter* no muestra un incremento significante con respecto a los resultados de la prueba 6.3.1, y esto puede deberse a las pérdidas que se encuentran distribuidas de forma uniforme. En la figura 6.14 se muestra como el número de paquetes recibidos, la tasa de Mbits/s o el tiempo de llegada presentan valores normales como en la figura 6.5 debido a que el porcentaje de pérdidas no es suficientemente alto como para alterar estos valores.

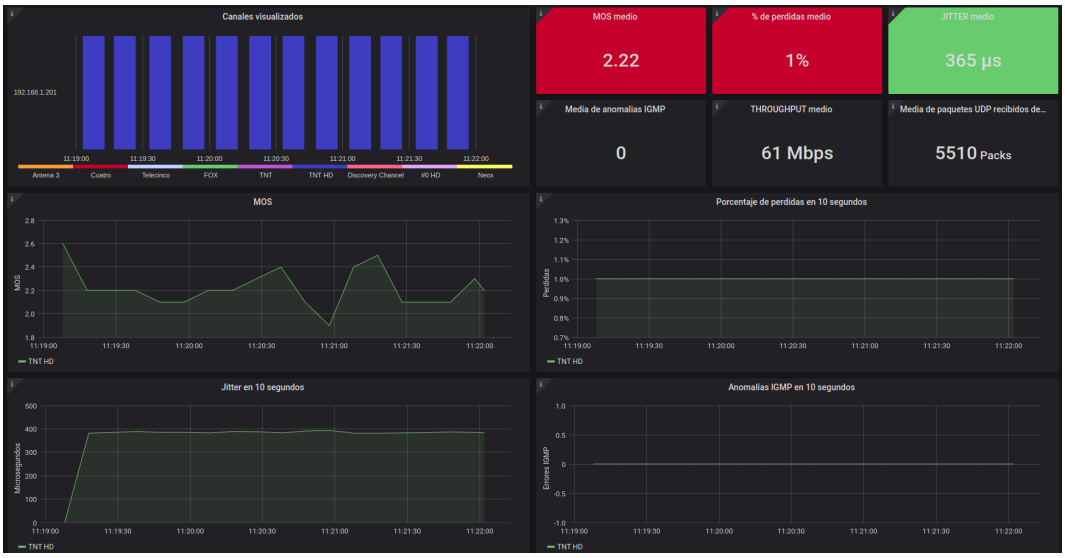


Figura 6.13: Gráficas de calidad de servicio generadas de una sesión IPTV con 1 % de pérdidas

Figura 6.14: Gráficas de información adicional generadas de una sesión IPTV con 1 % de pérdidas

6.4. Conclusiones

En este capítulo se ha explicado como se han obtenido las distintas trazas de red para realizar las pruebas y como nuestra aplicación funciona ante distintos los distintos escenarios.

A partir de las pruebas realizadas, se puede afirmar que la herramienta proporciona un análisis correcto de los sucesos que pueden ocurrir durante una sesión IPTV. En el análisis de las calidades de servicio se ha demostrado que la aplicación es capaz de calcular en tiempo real las pérdidas que ocurren en la sesión, anomalías que se produce en el protocolo IGMP y calcular el *jitter* a medida que la sesión avanza.

En el siguiente capítulo se resume las conclusiones que se han obtenido en el trabajo y los posibles trabajos futuros de la aplicación.

7

Conclusiones y trabajo futuro

7.1. Introducción

En este último capítulo se recogen las conclusiones de este documento donde se explican los resultados obtenidos, los conocimientos utilizados y el potencial del proyecto en futuros trabajos.

7.2. Conclusiones

El objetivo de nuestro TFG ha sido desarrollar una herramienta de monitorización de los servicios de multimedia sobre redes IP a través del análisis de la sesiones IPTV. Con este objetivo, se ha desarrollado una aplicación que detecta las peticiones de suscripción a los grupos de difusión *multicast* y a partir de ese momento se analiza la sesión IPTV y detecta los posibles errores que puedan surgir en la sesion. Por último, los resultados que se obtienen tras el análisis se muestran en forma de gráficas con la aplicación de Grafana.

Hemos observado que una herramienta que se encarga del análisis del tráfico en tiempo real necesita un tiempo de ejecución rápido debido a la gran cantidad de información que puede llegar en un segundo, por lo tanto se necesita un equipo hardware con presentaciones mínimas para poder llevar a cabo el análisis, y un lenguaje de programación rápido como C para analizar cada paquete lo más rápido posible.

Por un lado hemos comprobado que el tráfico TCP, externo a la sesión IPTV, no causa estragos en la sesión IPTV gracias a que el protocolo de TCP evita la congestión y a que IPTV reserva parte del ancho de banda del cliente para una comunicación privada entre el servidor multimedia y el cliente.

Por otro lado hemos visto que un tráfico externo UDP si puede afectar a la calidad de la sesión IPTV, debido a que el protocolo UDP no presenta controles de flujo como en TCP, causando pérdidas y aumentando el *jitter* de la sesión.

Por último, concluir que este trabajo ha sido posible gracias a varios conceptos que habían sido presentados en diferentes asignaturas del grado. A nivel de programación, Programación I y Progra-

mación II son las asignaturas que me enseñaron a programar en C y a utilizar las estructuras de datos que se han utilizado en la aplicación. Por otro lado, las asignaturas de Redes I, Redes II y Redes Multimedia son las asignaturas que me explicaron el funcionamiento de los protocolos y la programación a nivel de red. Con respecto a la base de datos, la asignatura de Estructura de Datos me enseñó a crear bases de datos y realizar consultas a estas, y el conocimiento previo que tenía de Grafana fue posible gracias a las prácticas externas que curse este mismo año.

7.3. Trabajo Futuro

En un futuro, nos gustaría implementar este tipo de aplicación en otros tipo de tecnologías como los servicio Over-The-Top (OTT) [21] en los que se encuentra Netflix o HBO, ya que son servicios que actualmente tienen una gran demanda y observar que diferencias se encuentran a nivel de calidad con los servicios IPTV.

Otro aspecto interesante sería probar nuestra aplicación en redes de alta velocidad para simular el escenario de una red troncal de un operador.

Otro idea interesante sería realizar aplicaciones similares a esta pero sobre otros servicios de IPTV como aplicaciones de vídeo interactivas o televisión en teléfonos móviles o tablets.

Bibliografía

[1] M. G. P. Fernando Boronat Seguí and J. L. Mauri, *IPTV, La televisión por internet*. Vértice, 2009 (Visto el 17/04/2020).

[2] A. A. Mahimkar, Z. Ge, A. Shaikh, J. Wang, J. Yates, Y. Zhang, and Q. Zhao, "Towards automated performance diagnosis in a large iptv network," *SIGCOMM Comput. Commun. Rev.*, 2009 (Visto el 09/05/2020).

[3] D. F. Q. Matilde Delgado, "Iptv: estructura de mercado y tipología de la oferta en españa," *ZER - Revista de Estudios de Comunicación*, 2006 (Visto el 11/05/2020).

[4] S. Shoaf and M. Bernstein, "Introduction to igmp for iptv networks," *Jupiter Network*, 2006 (Visto el 21/04/2020).

[5] Atresmedia Corporación, "Antena3." Disponible en `https://www.antena3.com`, 1988 (Visto el 12/05/2020).

[6] K. G. Thomas Schierl and T. Wiegand, "Introduction to igmp for iptv networks," *Fraunhofer HHI*, 2009 (Visto el 21/04/2020).

[7] P. Fouliras, "On rtp filtering for network traffic reduction," Association for Computing Machinery, 2008 (Visto el 19/05/2020).

[8] OCDE, *Estudio de la OCDE sobre telecomunicaciones y radiodifusión en México 2017*. Org. for Economic Cooperation Development, 2017 (Visto el 25/04/2020).

[9] Cisco, "Enterprise qos solution reference network design guide," 2014 (Visto el 04/05/2020). Disponible en `https://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/WAN_and_MAN/QoS_SRND/QoS-SRND-Book/QoSIntro.html`.

[10] Alberto Mozo Robles, "Diseño e implementación de un sistema de medición y monitorización de la calidad de servicio y experiencia de redes IPTV basadas en el códec H.264." 2016 (Visto el 23/03/2020).

[11] I. T. Union, "P.910 : Subjective video quality assessment methods for multimedia applications," 2008 (Visto el 10/12/2019).

[12] I. T. Union, "Definition of quality of experience (qoe)," 2007 (Visto el 13/05/2020).

[13] "Pcap4J." Disponible en `https://www.javadoc.io/doc/org.pcap4j/pcap4j/1.7.3/index.html`, 2011 (Visto el 21/05/2020).

[14] Gordon McMillan, "Socket Programming HOWTO." Disponible en `https://docs.python.org/3/howto/sockets.html`, 2001 (Visto el 21/05/2020).

[15] Tcpdump Group, "Tcpdump Libpcap." Disponible en `https://www.tcpdump.org`, 2010 (Visto el 21/05/2020).

[16] Mark Dreker, ostezer, "SQLite vs MySQL vs PostgreSQL: A Comparison Of Relational Database Management Systems." Disponi-

ble en https://www.digitalocean.com/community/tutorials/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-syst 2019 (Visto el 18/05/2020).

[17] Elastic NV, "Kibana." Disponible en https://www.elastic.co/es/kibana, 2019 (Visto el 16/05/2020).

[18] Raj Dutt, Torkel Ödegaard, Anthony Woods, "Grafana Labs." Disponible en https://grafana.com, 2014 (Visto el 30/01/2020).

[19] The Wireshark team, "Wireshark · Go Deep." Disponible en https://www.wireshark.org, 1999 (Visto el 21/05/2020).

[20] C.-C. Wen and C.-S. Wu, "Qos supported iptv service architecture over hybrid-tree-based explicit routed multicast network," *International Journal of Digital Multimedia Broadcasting*, 2012 (Visto el 01/05/2020).

[21] D. H. K. Seongcheol Kim, Hyunmi Baek, "Special issue: Ott and live streaming services: Past, present and future," *Telecommunications Policy*, (Visto el 18/05/2020).

[22] I. S. Institute, "Internet protocol," *INTERNET STANDARD*, 1981 (Visto el 27/04/2020). Disponible en https://tools.ietf.org/html/rfc791.

[23] R. F. H. Schulzrinne, S. Casner and V. Jacobson, "Rtp: A transport protocol for real-time applications," *INTERNET STANDARD*, 2003 (Visto el 23/04/2020).

[24] W. C. Fenner, "Internet group management protocol, version 2," *INTERNET STANDARD*, 1997 (Visto el 21/04/2020).

Definiciones

broadcast Transmisión de información desde un emisor a una multitud de receptores de forma simultánea.

jitter Variación del retardo de paquetes.

live television Televisión en vivo.

multicast Multidifusión.

QoE Calidad de experiencia.

QoS Calidad de servicio.

throughput Tasa de transferencia.

unicast Difusión única.

Acrónimos

DCCP Datagram Congestion Control Protocol.

IGMP Internet Group Management Protocol.

IPTV Internet Protocol Television.

ISP Internet Service Provider.

MOS Mean Opinion Score.

OTT Over-The-Top.

RAG Red Amber Green.

RTP Real-time Transport Protocol.

SHO Super Head-end Office.

STB Set-Top-Box.

TCP Transmission Control Protocol.

UDP User Datagram Protocol.

VHO Video Head-end Office.

VoD Video on Demand.

Apéndices

A

Fórmulas

Este apéndice muestra las fórmulas utilizadas en el calculo de las calidades del servicio y de la estimación del MOS.

$$Jitter = \sqrt{\frac{{Ret_n}^2 - {Ret_{n-1}}^2}{NumPaq_n - NumPaq_{n-1}} - \left(\frac{Ret_n - Ret_{n-1}}{NumPaq_n - NumPaq_{n-1}}\right)^2} \tag{A.1}$$

$$Retardo = \frac{\sum TiempoEntrePaquetes}{Paquetes} \tag{A.2}$$

$$Mbits/s = \frac{Bytes * 8}{10^6} \tag{A.3}$$

$$\%Pérdidas_T = \frac{Pérdidas_T}{Pérdidas_T + Paquetes_T} \tag{A.4}$$

$$\%Pérdidas = \frac{Pérdidas}{Pérdidas + Paquetes} \tag{A.5}$$

$$MOS = 1 + \frac{4}{\left(1 + \frac{Pérdidas}{2}\right)^3} \tag{A.6}$$

B

Cabeceras de protocolos

El siguiente apéndice muestra las cabeceras de los protocolos que han sido utilizados en el proyecto.

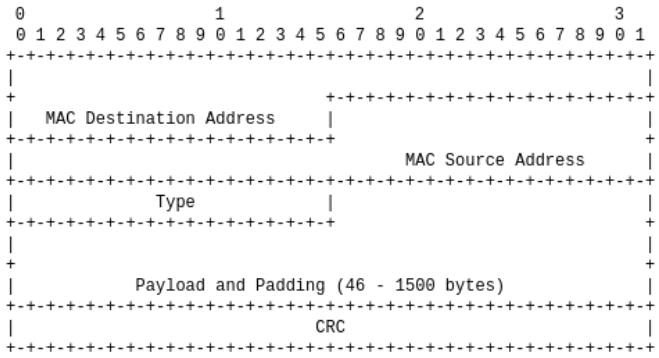


Figura B.1: Cabecera Ethernet

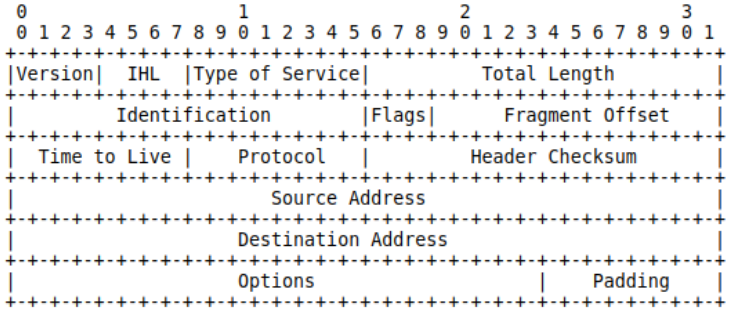


Figura B.2: Cabecera IP [22]

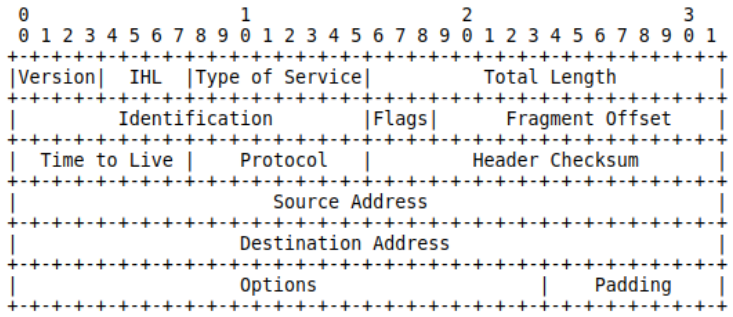


Figura B.3: Cabecera RTP [23]

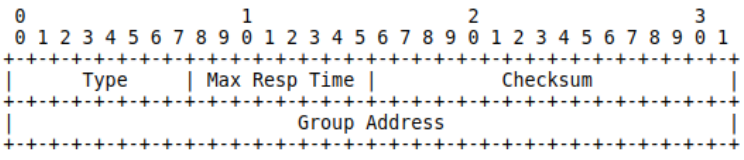


Figura B.4: Cabecera IGMPv2 [24]

C

Mensajes IGMP

El siguiente apéndice muestra los mensajes IGMP con los que se ha trabajado en el trabajo.

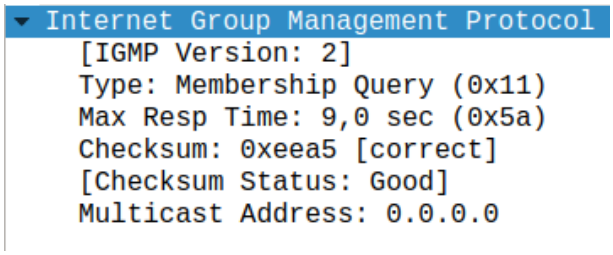


Figura C.1: Ejemplo *Membership Query*

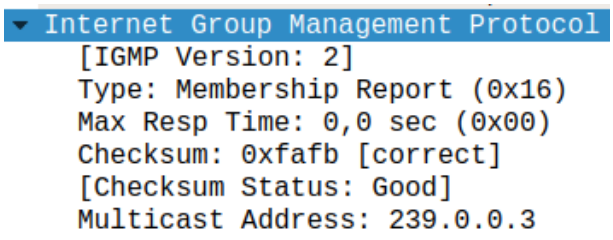


Figura C.2: Ejemplo *Membership Report*

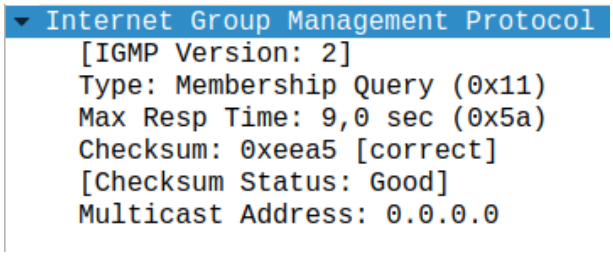


Figura C.3: Ejemplo *Leave Group*

D

Muestras de traza de las pruebas

El apéndice muestra parte del contenido de los ficheros de red utilizados en las pruebas desde Wireshark.

15167	11:19:22,353234	172.26.73.225	239.0.5.87	UDP	1386	60592 → 8208 Len=1344
15168	11:19:22,354630	172.26.73.225	239.0.5.87	UDP	1386	60592 → 8208 Len=1344
15169	11:19:22,356995	172.26.73.225	239.0.5.87	UDP	1386	60592 → 8208 Len=1344
15170	11:19:22,358600	172.26.73.225	239.0.5.87	UDP	1386	60592 → 8208 Len=1344
15171	11:19:22,360184	172.26.73.225	239.0.5.87	UDP	1386	60592 → 8208 Len=1344
15172	11:19:22,361581	172.26.73.225	239.0.5.87	UDP	1386	60592 → 8208 Len=1344
15173	11:19:22,362623	192.168.1.201	239.0.5.87	IGMPv2	60	Membership Report group 239.0.5.87
15174	11:19:22,363770	172.26.73.225	239.0.5.87	UDP	1386	60592 → 8208 Len=1344
15175	11:19:22,365322	172.26.73.225	239.0.5.87	UDP	1386	60592 → 8208 Len=1344
15176	11:19:22,366846	172.26.73.225	239.0.5.87	UDP	1386	60592 → 8208 Len=1344
15177	11:19:22,368366	172.26.73.225	239.0.5.87	UDP	1386	60592 → 8208 Len=1344
15178	11:19:22,369895	172.26.73.225	239.0.5.87	UDP	1386	60592 → 8208 Len=1344
15179	11:19:22,372238	172.26.73.225	239.0.5.87	UDP	1386	60592 → 8208 Len=1344
15180	11:19:22,373724	172.26.73.225	239.0.5.87	UDP	1386	60592 → 8208 Len=1344
15181	11:19:22,375227	172.26.73.225	239.0.5.87	UDP	1386	60592 → 8208 Len=1344
15182	11:19:22,376803	172.26.73.225	239.0.5.87	UDP	1386	60592 → 8208 Len=1344

Figura D.1: Muestra de traza de una sesión IPTV normal

76916	17:13:17,292476	172.26.74.9	239.0.0.4	UDP	1386	48940 → 8208 Len=1344
76917	17:13:17,295525	172.26.73.233	239.0.0.3	UDP	1386	55167 → 8208 Len=1344
76918	17:13:17,295756	192.168.1.40	239.0.0.3	IGMPv2	60	Membership Report group 239.0.0.3
76919	17:13:17,296952	172.26.74.9	239.0.0.4	UDP	1386	48940 → 8208 Len=1344
76920	17:13:17,299984	172.26.73.233	239.0.0.3	UDP	1386	55167 → 8208 Len=1344
76921	17:13:17,301495	172.26.74.9	239.0.0.4	UDP	1386	48940 → 8208 Len=1344
76922	17:13:17,305184	172.26.73.233	239.0.0.3	UDP	1386	55167 → 8208 Len=1344
76923	17:13:17,305978	172.26.74.9	239.0.0.4	UDP	1386	48940 → 8208 Len=1344
76924	17:13:17,309742	172.26.73.233	239.0.0.3	UDP	1386	55167 → 8208 Len=1344
76925	17:13:17,310520	172.26.74.9	239.0.0.4	UDP	1386	48940 → 8208 Len=1344
76926	17:13:17,314253	172.26.73.233	239.0.0.3	UDP	1386	55167 → 8208 Len=1344
76927	17:13:17,314994	172.26.74.9	239.0.0.4	UDP	1386	48940 → 8208 Len=1344
76928	17:13:17,318707	172.26.73.233	239.0.0.3	UDP	1386	55167 → 8208 Len=1344
76929	17:13:17,319454	172.26.74.9	239.0.0.4	UDP	1386	48940 → 8208 Len=1344
76930	17:13:17,323291	172.26.73.233	239.0.0.3	UDP	1386	55167 → 8208 Len=1344
76931	17:13:17,323988	172.26.74.9	239.0.0.4	UDP	1386	48940 → 8208 Len=1344

Figura D.2: Muestra de traza de una sesión IPTV normal con dos clientes

19010	16:59:43,545080	192.168.1.35	172.217.168.174	TCP	66	37270 → 443 [ACK] Seq=94448 Ack=119641 Win=6566 Len=0 TSval=4036446126 TSecr=3870944915
19011	16:59:43,545091	172.217.168.174	192.168.1.35	TLSv1.2	1556	Application Data, Application Data
19012	16:59:43,545096	192.168.1.35	172.217.168.174	TCP	66	37270 → 443 [ACK] Seq=94448 Ack=121131 Win=6556 Len=0 TSval=4036446126 TSecr=3870944915
19013	16:59:43,546360	172.26.76.10	239.0.0.74	UDP	1386	57266 → 8208 Len=1344
19014	16:59:43,550734	172.26.76.10	239.0.0.74	UDP	446	57266 → 8208 Len=404
19015	16:59:43,552355	172.26.76.10	239.0.0.74	UDP	1386	57266 → 8208 Len=1344
19016	16:59:43,556887	172.26.76.10	239.0.0.74	UDP	1386	57266 → 8208 Len=1344
19017	16:59:43,561577	172.26.76.10	239.0.0.74	UDP	1386	57266 → 8208 Len=1344
19018	16:59:43,565588	172.26.76.10	239.0.0.74	UDP	1386	57266 → 8208 Len=1344
19019	16:59:43,568699	192.168.1.35	74.125.168.40	TCP	66	33980 → 443 [ACK] Seq=1060 Ack=5772 Win=64128 Len=0 TSval=1061229407 TSecr=2030588480
19020	16:59:43,568708	192.168.1.35	74.125.168.40	TCP	66	33978 → 443 [ACK] Seq=1054 Ack=5772 Win=64128 Len=0 TSval=1061229407 TSecr=2030588479
19021	16:59:43,570614	172.26.76.10	239.0.0.74	UDP	634	57266 → 8208 Len=592
19022	16:59:43,572221	172.26.76.10	239.0.0.74	UDP	1386	57266 → 8208 Len=1344
19023	16:59:43,576796	172.26.76.10	239.0.0.74	UDP	1386	57266 → 8208 Len=1344
19024	16:59:43,581358	172.26.76.10	239.0.0.74	UDP	1386	57266 → 8208 Len=1344
19025	16:59:43,585267	172.26.76.10	239.0.0.74	UDP	1386	57266 → 8208 Len=1344

Figura D.3: Muestra de traza de una sesión IPTV con tráfico TCP

230	18:27:53,221041	192.168.1.40	255.255.255.255	UDP	60	33728 → 8080 Len=17
231	18:27:53,221042	192.168.1.40	255.255.255.255	UDP	60	33728 → 8080 Len=17
232	18:27:53,221081	192.168.1.40	255.255.255.255	UDP	60	33728 → 8080 Len=17
233	18:27:53,221081	192.168.1.40	255.255.255.255	UDP	60	33728 → 8080 Len=17
234	18:27:53,221081	192.168.1.40	255.255.255.255	UDP	60	33728 → 8080 Len=17
235	18:27:53,221081	192.168.1.40	255.255.255.255	UDP	60	33728 → 8080 Len=17
236	18:27:53,221081	172.26.76.42	239.0.0.107	UDP	1386	58848 → 8208 Len=1344
237	18:27:53,221082	192.168.1.40	255.255.255.255	UDP	60	33728 → 8080 Len=17
238	18:27:53,221082	192.168.1.40	255.255.255.255	UDP	60	33728 → 8080 Len=17
239	18:27:53,221082	192.168.1.40	255.255.255.255	UDP	60	33728 → 8080 Len=17
240	18:27:53,221083	192.168.1.40	255.255.255.255	UDP	60	33728 → 8080 Len=17
241	18:27:53,221083	192.168.1.40	255.255.255.255	UDP	60	33728 → 8080 Len=17
242	18:27:53,221142	192.168.1.40	255.255.255.255	UDP	60	33728 → 8080 Len=17
243	18:27:53,221142	192.168.1.40	255.255.255.255	UDP	60	33728 → 8080 Len=17
244	18:27:53,221142	192.168.1.40	255.255.255.255	UDP	60	33728 → 8080 Len=17
245	18:27:53,221142	192.168.1.40	255.255.255.255	UDP	60	33728 → 8080 Len=17
246	18:27:53,221143	192.168.1.40	255.255.255.255	UDP	60	33728 → 8080 Len=17

Figura D.4: Muestra de traza de una sesión IPTV con tráfico UDP

47528	11:20:07,932324	172.26.73.225	239.0.5.87	UDP	1386	60592 → 8208 Len=1344
47529	11:20:07,933923	172.26.73.225	239.0.5.87	UDP	1386	60592 → 8208 Len=1344
47530	11:20:07,935440	172.26.73.225	239.0.5.87	UDP	1386	60592 → 8208 Len=1344
47531	11:20:07,937606	172.26.73.225	239.0.5.87	UDP	1386	60592 → 8208 Len=1344
47532	11:20:07,938976	172.26.73.225	239.0.5.87	UDP	1386	60592 → 8208 Len=1344
47533	11:20:07,940634	172.26.73.225	239.0.5.87	UDP	1386	60592 → 8208 Len=1344
47534	11:20:07,942021	172.26.73.225	239.0.5.87	UDP	1386	60592 → 8208 Len=1344
47535	11:20:07,944273	172.26.73.225	239.0.5.87	UDP	1386	60592 → 8208 Len=1344
47585	11:20:52,616321	172.26.73.225	239.0.5.87	UDP	1386	60592 → 8208 Len=1344
47586	11:20:52,617840	172.26.73.225	239.0.5.87	UDP	1386	60592 → 8208 Len=1344
47587	11:20:52,619410	172.26.73.225	239.0.5.87	UDP	1386	60592 → 8208 Len=1344
47589	11:20:52,621565	172.26.73.225	239.0.5.87	UDP	1386	60592 → 8208 Len=1344
47590	11:20:52,622993	172.26.73.225	239.0.5.87	UDP	1386	60592 → 8208 Len=1344
47591	11:20:52,624522	172.26.73.225	239.0.5.87	UDP	1386	60592 → 8208 Len=1344
47592	11:20:52,626159	172.26.73.225	239.0.5.87	UDP	1386	60592 → 8208 Len=1344
47593	11:20:52,627022	172.26.20.39	239.0.2.2	UDP	123	40175 → 22224 Len=81

Figura D.5: Muestra de traza de una sesión IPTV con anomalías IGMP

Packet	Sequence	Delta (ms)	Jitter (ms)	Skew	Bandwidth	Marker	Status
104431	53926	9.81	0.28	-0.27	951139.58		Wrong sequence number
98567	47872	8.53	0.31	-0.27	891119.36		Wrong sequence number
85069	36498	8.50	0.29	-0.40	778044.83		Wrong sequence number
44951	1424	8.35	0.21	-0.44	429904.51		Wrong sequence number
69447	23408	7.51	0.34	-0.14	648438.27		Wrong sequence number
39879	62029	6.99	0.27	-0.79	381044.64		Wrong sequence number
67608	21866	6.93	0.28	-0.76	633039.97		Wrong sequence number
27730	50660	6.92	0.27	-0.53	268240.32		Wrong sequence number
16777	40441	6.92	0.32	-0.38	166480.19		Wrong sequence number
113855	62767	6.89	0.26	-0.40	1038756.80		Wrong sequence number
2302	26039	6.89	0.26	-0.66	23480.83		Wrong sequence number
96079	45678	6.88	0.24	-0.24	869341.02		Wrong sequence number
70941	24654	6.87	0.24	-0.58	660796.77		Wrong sequence number
108769	58381	6.87	0.26	-0.32	995060.10		Wrong sequence number
29642	52630	6.82	0.27	-0.43	287800.61		Wrong sequence number
55373	10683	6.82	0.26	-0.37	521583.46		Wrong sequence number

Figura D.6: Muestra de traza de una sesión IPTV con pérdidas

E

Modelo entidad-relación normalizado

La figura E.1 muestra el modelo entidad-relación normalizado. Como se explicó anteriormente, debido a que la normalización podría afectar al rendimiento de las consultas y a la poca información que se iba a utilizar se decidió prescindir de este modelo.

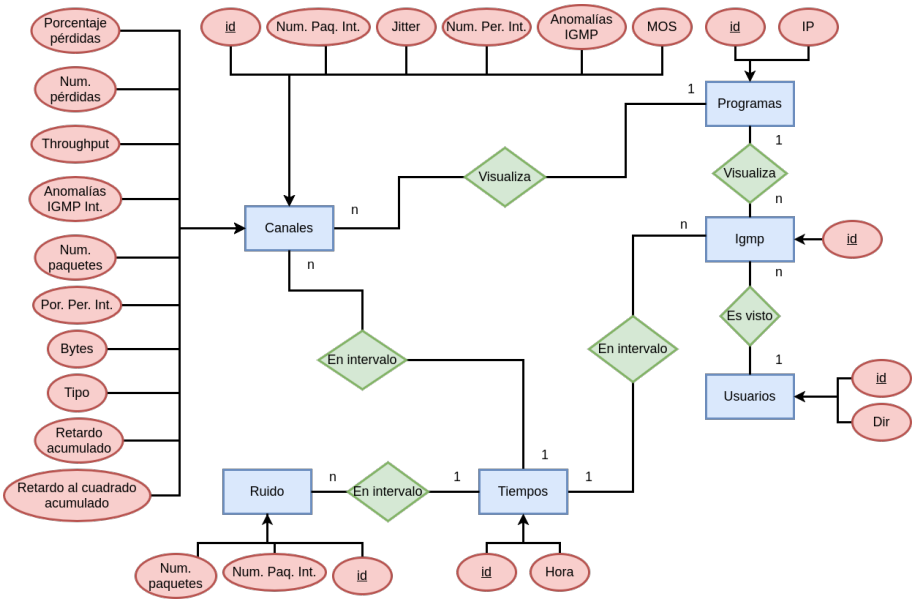


Figura E.1: Modelo Entidad-Relación normalizado

F

Repositorio GitHub

El código del proyecto se encuentra en el siguiente repositorio GitHub: **github/Lexiwi/Tfg**.

Dentro del repositorio se encuentra los .c y .h junto con el makefile para la compilación del programa. Además, se incluye el .json que genera Grafana para la creación de las gráficas.

Los ficheros de red utilizados en las pruebas del proyecto pueden descargarse en el siguiente enlace: **dropbox.com/ficherosPrueba**